UNIVERSITE CATHOLIQUE DE LOUVAIN

LOUVAIN SCHOOL OF MANAGEMENT

THE RISE OF DIGITAL ECOSYSTEMS: MULTIDIMENSIONAL AND CONTEXT-AWARE ADAPTATION OF USER INTERFACES

Promoteur : Jean Vanderdonckt

Mémoire-recherche présenté par
Denis Cariat

en vue de l'obtention du titre de
Master en ingénieur de gestion

ANNEE ACADEMIQUE 2012 -2013

Foreword

I would like to thank Jean Vanderdonckt (thesis supervisor) and his assistant Vivian Genaro Motti for her help.

I would like to thank my family for their support during my academic journey

Table of contents

II.

IV.

Introduction

Behind this computerized title is hidden a new trend in information systems' and user interfaces' conception which is more human and social than it appears by taking into account multiple context's dimensions such as the platform used, users' characteristics and specific environmental factors. This approach finds its roots in the emergence of new devices, such as smartphones and tablets that set information systems in motion, away from homes' and offices' desktop computers and creating a real digital ecosystem. Figures are remarkable: while 1,746 billion cell phones were sold in 2012, tablets are expecting to surpass desktop and laptop computers in 2017 with respectively 467 million units and 271 million units sold (GARTNER, 2013). Initially desktop computer-based, the Internet is then currently becoming predominantly mobile. However, numerous websites and information systems are still designed on this old stationary paradigm and is not adapted to this massive current trend, requiring cross-platform harmonization as well as design and interaction adaptations in order to fulfill both developers' and users' aspirations. Moreover, as a first reaction towards this new trend, the same website is often developed several times on different platforms and it induces important waste of money and resources. As a result, the wide adoption of new interconnected devices – mobile or stationary - used in different contexts adds some dimensions that need to be addressed in order to improve users' experience which is currently disappointing on mobile devices. For instance, the loading time is the number one frustration for mobile users, followed by formatting issues (KEYNOTE, 2012). The justification of this thesis and what makes multidimensional context-aware adaptations an interesting subject of research is that this area "is strongly driven by innovation, characterized by rapidly evolving use, and has enormous market potential and growth" (KJELDSKOV, 2013).

By embracing a holistic and multidimensional approach, the aim is a deep evaluation of the evolving Internet framework in order to consider all its aspects during the information systems' conception: from insights into past and emerging technologies to recommendations and guidelines to implement multidimensional context-aware adaptations. The focus is set on three context's dimensions (i.e. platform, user, environment) and related adaptations techniques selected in the Serenoa project documentation (the project is described in appendix 9). For this purpose, the methodology (described in the second part) consists in adaptation techniques within the Serenoa Platform as well as implementing and testing them on a website built for the occasion. This website is based on the Serenoa project's study case: a car rental website (see appendix 7). The scope of the analysis will be limited to websites' main features, user interfaces and interaction designs as they have the largest impact on users' experience which is the final purpose for developers. The thesis will provide a solid and useful basis for further researches on more complex information systems. Finally, this thesis does not claim to be the answer but claims more to be a web development tool including a set of previous researches' analysis, current practices, personal tests and personal recommendations for multidimensional context-aware adaptations' implementation and users' experience improvement. Here below are described the content and structure of the following parts and chapters. The thesis is divided into two parts. On the one hand, a theoretical part gathers qualitative as well as quantitative information. On the other hand, a practical part provides the methodology, analyzes previous researches, tests and evaluates specific adaptation techniques for each context's dimension. Final HTML, CSS and javascript files are available on the CD attached to this thesis.

In the first part, the first chapter focuses on the 'platform dimension' and introduces the mobile computing history which triggered implicitly the multidimensional approach with the first laptop computer released in 1981. Since then, breakthrough technologies have been developed and continuous improvements in interaction design have been made at an alarming rate through several waves of development. As a result, mobile computing history has just started its fourth decade but achievements are already remarkable: today's market is flooded by countless different models of user-friendly laptops, smartphones and tablets. In parallel, millions of software solutions and applications have been created to respond to users' needs. Having evolved on different parallel paths for several decades, desktop computers' and those new mobile devices' capabilities have now met thanks to advanced hardware development and the rapid evolution of wireless data networks. Nowadays, these technologies are complementary in users' everyday life and must be considered as parts of an ecosystem in which devices, systems, services as well as users and their surrounding environment are interacting entities that have to be studied from a holistic point of view. Finally, this chapter introduces the mobile interaction design field of research and the challenges that are addressed in this thesis.

Nevertheless, even though these technologies are at the cutting edge of the scientific and technologic research and are already widely spread worldwide, the second chapter describes how these new technologies coexist between them and are modifying drastically users' behaviors and the global economy. Firstly, insights into the digital ecosystem and trends for each of its components are given: desktop computers, laptops computers, smartphones and tablets as well as their operating systems, their browsers, their screen resolution, their Internet access points and their respective applications. Statistics also offer a large picture of users' – current and upcoming – trends in their uses and needs. Secondly, four researches' and surveys' key findings have been gathered and show that end-users are currently disappointed by the experience provided while browsing with their mobile devices. It highlights for instance that, in general, mobile users expect the loading time to be at least similar on mobile devices than on stationary devices. Without entering yet into details, it is also easily understandable that this mobile wave induces commercial and financial issues. On the one hand, commercial implications of this multi-device landscape are numerous, mainly for advertisements but it is also a new and complementary tool to develop a business, its turnover and its brand awareness. On the other hand, developing and maintaining a single information system on multiple separate platforms induces consequent amount of time and money. Many organizations see current technologies separately instead of seeing them as a whole. As a matter of fact, websites are often developed separately on different existing devices and this approach makes it difficult when it comes to update the content or when a new device comes out. Hence the need of technical harmonization, adaptations techniques as well as the need of an accurate understanding of the Internet's evolving structure and functioning. The lack of harmonization and thus the lack of multiplatform portability is the essence of the coming challenges. All these information and statistical data will be summarized in order to produce interaction design and adaptations' recommendations based on users' needs and experiences. Finally, this chapter also highlights the motivation and efforts made by the World Wide Web Consortium (W3C) to create and adopt standards within the Internet industry in order to harmonize the user experience on different platforms and different browsers.

The two first chapters focus on the 'platform' dimension by describing devices and technologies that are part of the global digital ecosystem as well as emerging users' behaviors towards these new technologies. However, with the wider adoption of mobile devices, other additional context's dimensions have to be addressed such as the user and its surrounding environment. In this purpose, the third chapter firstly introduces the primordial role of the context and definitions of *multidimensional adaptations* given by pioneers in informatics, researchers and developers in that specific field. Within these adaptations dimensions, challenges also rise. For instance, real-time adaptations to a wide range of environmental factors – which change rapidly, if not continuously – can lead to system thrashing. The uniqueness of each user would also produce huge amounts of data but this issue can be solved by using users' categorization. This chapter closes the first part of this thesis.

In the second and practical part, the methodology is first presented. Then, the fourth chapter introduces the three dimensions that will be studied thanks to previous theoretical researches: the platform, the environment and the user. Corresponding questions and challenges to be addressed are mentioned. This chapter also focuses on a suggested order of implementation for these dimensions. Indeed, cross-platform adaptations seem to be what users want the most before any other things. Moreover, it makes sense to adapt the layout across platforms first rather than environment's or users' adaptations that would not be properly displayed on different platforms. Subsequently, the 'environment dimension' may be implemented as a location is less specific than users' uniqueness. Of course, dimensions are complementary and the development is not perfectly serial but this suggested order makes the development clearer and straightforward.

The fifth chapter presents, tests and evaluates selected adaptations techniques for each dimension identified. While cross-platforms' adaptations become slowly a standard within the developers' community thanks to the broader adoption of HTML5, environment's adaptations are for the moment widely implemented and users' adaptations are still in their infancy. Tables sum up adaptations techniques for each dimension at the end of each section.

For cross-platform harmonization, developers' community bases their work on three concepts that are defined and explained: the 'responsive web design', the 'mobile first approach' and the 'progressive enhancement'. Afterwards, different related adaptations' techniques are implemented, tested and illustrated with the car rental website. Techniques consist in combining Media Queries and proportional dimensions instead of fixed ones. While Media Queries set thresholds (e.g. screen's width) in which contents' CSS rules can be displayed differently, proportional layouts allow contents to fit the screen's width between two Media Queries. Unfortunately, absolute screen's size (e.g. centimeters or inches) are not reliable and developers therefore have to use other metrics to use Media Queries properly. Furthermore, interaction designs underlying the mobile first approach are presented. Finally, as mobile devices have lower capabilities, it is crucial to adapt images' size and optimize the loading time which is currently disappointing mobile users. For that purpose, different techniques are presented and tested to adapt images across platforms. Programming best practices are listed and must be used from the beginning to not be time consuming afterwards. Gain from programming best practices are also computed on the car rental website.

The fast data networks' development and the broad adoption of mobile devices with embedded-GPS have triggered the implementation of a plethora of location-based services as well as the creation of multiple applications with location-based features. At first, this section explains how users' location may be retrieved by different means under the assumption that users give their agreement. Besides providing users' location, other environment-based adaptations can be implemented thanks to embedded sensors such as brightness regulation or surroundings' environment recognition thanks to embedded microphones. Further, different techniques to capture users' language are also detailed with respective pros and cons. Best practices in that domain are provided. Finally, the last section provides insights and figures allowing the comprehension of these location-based features' success. Indeed, figures state that the closer users are to a business, the more likely they are to click on a mobile advertisement for that specific business.

The user dimension is seen as a final feat by developers. For decades, researchers in computing and related fields have dreamed about devices and applications that adapt themselves to the user instead of the opposite. In order to improve users' experience, common sight's troubles are addressed and adaptations techniques (e.g. font types, font sizes, contrast) as well as examples are provided and tested for each of them. Secondly, less common troubles are also addressed such as the color-blindness, the blindness and the hands' tremor. Thirdly, insights are provided into techniques allowing the detection of users' current activity (e.g. walking, running, biking). Finally, examples of users' personality detection and categorization are provided through an example of Safety Driving application.

As a conclusion, the last chapter provides a features' diagram and related tables as a summary which illustrates and explains in which manner selected adaptations techniques can be combined or not. Both developers and users have decisions to take while respectively implementing or using a website. On the hand, it provides developers with a web development tool in which they select adaptations techniques according to the website's purpose and their resources (e.g. time, money). On the other hand, it allows users to enable or disable device/browser features according to their adaptation expectations and their privacy concerns. Furthermore, advantages and shortcomings of this analysis are presented. Discussions will also be held on several topics such as the relevance of specific adaptations techniques in comparison with their development costs and length as well as rising privacy issues. From those shortcomings, the path for further researches is drawn.

10.

First part: bibliographic research, quantitative research and adaptation dimensions identification

CHAPTER I: The Mobile Computing History in several waves

As previously presented in the introduction, the multidimensional approach has emerged with the mass production of smartphones and tablets (or even PDAs). While studying these devices and their use, we therefore have to start speaking about their origin: the Mobile Computing research field. As explained in the next section, this three-decades-old field of research was technical-oriented at its starting point but nowadays, researches are deeply oriented on areas such as interaction design, user experience and usability. The following quote makes clear how researches have been conducted and why this has been so fast: “The field of mobile computing has its origin in a fortunate alignment of interests by technologists and consumers. Since the dawn of the computing age, there have always been technological aspirations to make computing hardware smaller, and ever since computers became widely accessible, there has been a huge interest from consumers in being able to bring them with you” (ATKINSON, 2005).

This history focuses on the ‘platform dimension’ and does not highlight every model updates but emphasizes on specific milestones that provided radical improvements and changed users’ everyday lives. Afterwards, definitions and challenges for mobile interaction design will be presented.

This is an appropriate starting point to set the scene and explore multidimensional adaptations. Other dimensions will be presented in further chapters.

1.1. Waves of development

Mobile computing researches can be divided into seven waves. Nevertheless, these waves have not been either perfectly sequential or perfectly simultaneous but they have been focusing on specific trends that have motivated deep researches. They are very helpful to see how interaction design for mobile devices has become such an important field of research today.

The ‘*Portability’* wave focused on size reduction to enable the production of computers that users can bring with them easily. During the ‘*Miniaturization’* wave, researches targeted the creation of smaller devices that users could use on the move.

The '*Connectivity'* wave saw the emergence of mobile devices and applications that allowed Internet browsing and communication through wireless data networks. On the one hand, the next wave, '*Convergence'* gave birth to all-in-one devices, gathering in the same device a plethora of specialized mobile devices such as music players, Personal Digital Assistants (PDAs), cell phones, games and cameras. On the other hand, the '*Divergence'* wave promoted a multi-specialized-devices approach arguing that all-in-one devices provide average performance in everything they achieve.

Finally, users are currently experiencing the sixth and seventh waves, respectively the '*Applications'* wave and the '*Digital Ecosystems'* wave. The first one is about promoting consumption by offering users content and substance as well as functional, enjoyable and user-friendly applications. Last but not least, the wave of digital ecosystems is characterized by its pervasive aspect which is stirring up users' everyday life. This concerns "larger wholes of pervasive and interrelated technologies that interactive mobile systems are increasingly becoming a part of" (KJELDSKOV, 2013). A predominant factor is the cloud computing that allows users to access their data from wherever they want and from whatever device they use (stationary or mobile). This has led to the creation of real ecosystems and that we need to understand, create and maintain.

This chapter focuses more on trends than figures. Then, in order to illustrate the scope of these two last waves and the coming trends, relevant figures and statistics are provided in the second chapter. The two first waves (i.e. portability and miniaturization) are available in appendix 4 as the connectivity wave is the real starting point for the success of mobile devices.

1.1.1. Connectivity

Not considered yet to be computers, the first handheld – mobile – phone was commercialized in 1983, almost ten years before the first PDA Apple Newton (in 1992): the Motorola DynaTAC 8000X (see appendix 1, figure 4). Later, in 1991, the Global System for Mobile Communications (GSM) and the introduction of the Short Message Service (SMS) triggered the virtuous development of more complex and more functional mobile phones to reach sophisticated mobile devices of today.

In the 1990s, Nokia was with no doubt the leader in the interaction design researches for mobile devices. Limitations were numerous as they had to deal with small low-resolution displays and keyboard was limited to twelve numeric buttons and few navigation buttons.

Nevertheless, researchers achieved great improvements and reached commercial success with several devices such as the Nokia 3110 (see appendix 1, figure 5). “It introduced a simple graphical menu system and the ‘Navi-key’ concept for simplifying user interaction (i.e. an interaction design that reached the hands of more than three hundred million users through subsequent Nokia handsets). The basic interaction design of the Nokia 3110 was extended with T9 predictive text for SMS messaging, games, customizable ring tones, and changeable covers for the extremely successful Nokia 3210” (LINDHOLM & KEINONEN, 2003). These versions were therefore pioneers in the mass production of devices both designed for an intuitive interaction and easy customization.

This success and the growing use of SMS led to a first attempt to integrate the Internet into mobile phones: the WAP or Wireless Application Protocol. However, because of its poor usability and its slow data transfer, WAP never met the expectations (RAMSAY & NIELSEN, 2000). Later, more advanced mobile devices will fix this issue by accessing the real web.

1.1.2. Convergence

This fourth mobile computing wave was one of the most interesting as researchers started combining different types of specialized mobile devices such as PDAs and mobiles phones. They transformed them into “new types of hybrid devices with fundamentally different form factors and interaction designs” (KJELDSKOV, 2013). This first phase of convergence led to a large collection of innovative forms and interaction designs. Notable improvements are the following: adjustable device shape according to specific tasks (e.g. calling, typing), introduction of wider mobile phones with full keyboard (such as first Blackberry versions, see appendix 1, figure 10) and in 1992 was released the first full touch-screen mobile phone: the IBM Simon (see appendix 1, figure 6). There were no buttons and everything was enabled by using fingers or a pen.

Whereas the first phase of convergence attracted mostly professionals, a second wave appeared and was more attractive for the rest of the population as these devices included multiple media capabilities allowing users to listen to radio stations, to record and watch videos, to take pictures, to play music or to watch television. This second wave of multimedia phones has definitely inspired today’s smartphones that people use for leisure as well as for socializing or working.

In 2009, almost two billion camera phones were produced and through new social networks such as Facebook, allowing users to capture and share pictures, this new kind of photography had a huge social impact. However, “whereas early camera phones were clearly phones with cameras, novel interaction design led to several converged devices truly blurring the boundaries between the two” (MURPHY et al., 2005). A good example of this ambiguity is the Nokia N90 released in 2005 (see appendix 1, figure 7). Other devices focused on different capabilities such as listening to music (Sony Ericsson W600 in 2005, see figure 10 in appendix 1) or even playing video games (Nokia N-Gage in 2003, see figure 13 in appendix 1). All these devices went one step further in the interaction design area by offering users unprecedented experiences.

“The fundamental driver behind the trend of convergence is that mobile user experience is proportionally related to the functional scope of interactive mobile devices and systems: more means more” (MURPHY et al. 2005). As a result, these technologies have been often criticized and have been compared to a Swiss knife. Indeed, these devices were offering “clumsy technology with a wide range of functions, none of which are ideal in isolation” (NORMAN 1998, BERGMAN 2000, BUXTON 2001).

From another point of view, strength of convergence does not lie on the availability of numerous embedded –weaker – technologies but lies rather in additional opportunities of unprecedented interactions. Good examples are: taking pictures and directly sharing it on social networks, browsing the Internet or purchasing and directly listening songs on iPods.

1.1.3. Divergence

In opposition to the *Convergence* wave, “the fundamental view promoted by the trend of *Divergence* is that mobile user experience is inversely proportionate to the functional scope of interactive mobile devices and systems: less is more” (MURPHY et al., 2005). In other words, it is preferable to own several specialized devices than all-in-one devices that provide average – if not low – performance.

As their predecessors such as PDAs, mobile phones or GPS, these latest specialized devices were apparently nothing new but allowed the enhancement of well-known paradigms of use. Therefore, without being a technical necessity, the 2000s saw the emergence of deliberate and advanced interaction design choices. Without a doubt, the icon of this *Divergence* wave is the iPod released in 2001 by Apple (see appendix 1, figure 14). “Although not the first mobile digital music player, its interaction design, including the integration with iTunes and later the iTunes Music Store, fundamentally changed global music consumption and purchasing behavior" (KJELDSKOV, 2013). Even if many other mobile devices integrated an MP3 player, the iPod were preferred by far for its interaction design, its unprecedented user experience and its fashionable design. At the end of 2010, sales reached 290 million units. This success is still going on with further updates and adapted versions such as the iPod Nano released in 2010.

Another emblem of this wave is the video game console Sony PSP released in 2004 (see appendix 1, figure 11).

"The interaction design challenge of a diverged mobile device is considerably different from that of a converged one because its functional scope is much narrower. However, as diverged devices are by definition typically used in concert with a plethora of other interactive devices and systems unknown to the designer, there is a huge interaction design challenge in supporting good and flexible integration and 'convergence-in-use' "(MURPHY et al., 2005).

1.1.4. Applications

In 2007, a revolution in the interaction design field arose: the release of the iPhone (see appendix 1, figure 9) on the market by Apple. This was a converged device as many built before but what distinguishes it from its predecessors is the complete rethinking and the reshape of previous mobile interaction designs with remarkable interaction features improvements and choices. Alongside innovative features such as more fluid and aesthetic menus or the high-resolution and multi-touch display integrating gestures recognition (e.g. swiping, pinching) which definitely made keyboards and pens outdated, ground-breaking evolutions came from new-born "context-aware" capabilities. Indeed, thanks to integrated sensors and devices, the display depends on the device orientation (e.g. landscape, portrait, call mode). In its next versions, the embedded GPS went further in the "context-awareness", enabling location-based services. These capabilities were the first actual and practical steps in the multidimensional context-aware area and opened the door for further researches and features developments.

User experience and interaction designs have also been improved on embedded software. As a result, browsing, and emailing became much easier to such an extent that many users prefer using their smartphones rather than their computer for these specific tasks. In addition to those features, dedicated applications appeared soon such as Youtube or the iTune Store allowing users to respectively watch videos and purchase songs. "In concert, this meant that people actually started using their mobile device as a preferred gateway to the Internet, rather than as a last resort” (KJELDSKOV, 2013).

With more than 15 million sales during its first year, the Apple iPad (see appendix 1, figure 12) met an unprecedented and unexpected success once released in April 2010. Until then, tablets were a blurred category somewhere between computers and smartphones. However, with meticulous researches on interaction designs, Apple transformed it into a mobile device that is more than simply a computer without keyboard. This led to the exploration of the iPad form factor and to the development of web content as well as native applications especially designed for iPads. Until then, this had remained unexplored (CHEN, 2010) but Apple turned tablets into one of the most promising and interesting mobile device on earth.

Between iPhones and iPads, the biggest revolution came in 2008 with the launch of the online ‘App Store’ that offered users an unprecedented plethora of third-party applications for their iPhones – and iPads since 2010 –. From productivity tools to games, these applications are easy to download, install and use, no matter users’ computing skills and experience. Only six years after the first Apple iPhone and three years after the first iPad, figures are remarkable (KAMEKA, 2013): 900.000 active applications, 375.000 active applications especially developed for iPads, 50 billion applications downloaded (for iPhone, iPad and iTouch devices) since 2008 and 6 million registered iOS developers, among them 1.5 million signed in last year.

With the iPhone and the iPad, Apple totally reshaped mobile computing and set new frameworks as well as new standards for user experiences and mobile interaction designs. What they achieved is truly remarkable, such that the main competitor, Google took more than four years to catch up, with its own open source mobile operating system Android and its associated applications market (i.e. Google Play). Nowadays, these two giants are almost equal in term of applications downloads: around fifty billion applications downloaded but Google has already overtaken Apple concerning recent monthly downloads (2 billion for Google against 1.8 for Apple). Nevertheless, Apple still got the lion’s share when it comes to revenues generated even if Google is catching up with a significant market share increase from 19% to 27% between January and June 2013 (HALLECK, 2013). More details will be provided in the second chapter.

1.1.5. Digital Ecosystems

The integration by the population of these mobile devices in its everyday life in combination with the recent uptake for the cloud computing announces an imminent shift away from desktop computers that is already in progress. Indeed, as depicted with figures in the second chapter, mobile browsing will soon dominate the Internet. These elements are illustrating that the march is on towards the creation of broad digital ecosystems in which mobile and stationary computer systems will be part of. In order to understand, create and maintain this kind of ecosystems, it will be required considering them from a holistic angle, as any other ecosystem, and solve new interaction design challenges.

1.2. Definitions of mobile interaction design

Used for the first time in the 1980s by Bill Moggridge and Bill Verplank, interaction design is defined as “designing interactive products to support the way people communicate and interact in their everyday and working lives” (SHARP et al. 2007, p. 8). A broader definition of interaction design is “the design of everything that is both digital and interactive with particular attention to its subjective and qualitative aspects” (MOGGRIDGE, 2007, p. 660). “In other words, it is about creating life and work enhancing user experiences through the design, development, construction, and implementation of interactive products, devices, systems, and services” (KJELDSKOV, 2013).

Previous sections explained how previous interaction design challenges have been solved and how this field of research became crucial since the emergence of mobile devices. Moreover, the craze for Apple products and current similar devices – more than 200 million smartphones and 52.5 million tablets have been sold during the fourth quarter of 2012 (GARTNER REPORT, 2013) – confirmed speculations made by interaction design researchers about what users want to achieve with their mobile devices. This field of research surpassed former fields of research which are still focusing on the hardware side such as the battery lifetime, the network speed, input and output capabilities or the applications development. Today, these areas are less an issue than before and in the context of this thesis, the majority of these problems can be considered as solved. So, what is the next step? What are the next challenges for interaction design? Previous history showed that is quite unlikely that development has reached an end point.

1.3. Multidimensional adaptation and Interaction design challenges

Nowadays, we are therefore actually facing a practical problem as the majority of the population does not see the Internet as an ecosystem. There is a significant discrepancy between the way stakeholders perceive and use the Internet and the way the Internet is evolving and structured, even thought the majority still ignores this problem. A huge majority sees the Internet as it has been mainly spread over the world, static and desktop computer based while users nowadays access the internet at home, at work, in their car, while commuting, running or cycling and from a multitude of different devices (i.e. mobile or stationary). As a result, many websites and information systems are designed on this ‘not that old’ stationary paradigm and is not adapted to this massive current trend, requiring interaction design and features adaptations in order to fulfill both developers’ and users’ aspirations. Therefore, “the challenge of designing mobile interactions is going to be about creating interactive systems, services and applications that respond to the broad and diverse aspects of human life, and these not only provide utility and are easy to use, but also provide pleasure and fit naturally into peoples’ complex and dynamic lives of constantly changing settings and situations” (KJELDSKOV, 2013). Indeed, everything indicates a focus on content, applications, adaptations and interaction designs development as no major improvements have been made related to the physical form of recent devices. These forms seem to have reached stability, at least for a while.

CHAPTER II: Digital Ecosystems: description and figures

Until now, remarkable achievements have been made in many fields which lead our society to live in a real digital ecosystem gathering various stationary and mobile devices, a multitude of Internet browsers and a plethora of applications and software. By analogy with the state of the 'world wide web' in the 1990's, contemporary researchers are constantly pushing technological boundaries and all the figures depicted in this section indicate that everything has just started. Indeed, "there is a lot of excitement and interest, the development tools are easily accessible, and there is a huge audience of potential users" (KJELDSKOV, 2013).

Firstly, this chapter provides detailed descriptions and figures describing the digital ecosystem in which users are living in and the repartition between each of the three main platforms (i.e. smartphones, tablets and desktops/laptops). Details about various web browsers, screen resolutions, Internet traffic and access points as well as applications markets are detailed in appendix 8. However, key findings are provided in the section 2.1.2 Key Findings.

Secondly, data from accurate surveys and studies will introduce which issues users are currently facing and have to be addressed in order to harmonize and increase users experience whoever and wherever they are as well as whatever they use in this digital ecosystem.

Finally, the last section introduces the World Wide Web Consortium which works for the progressive development of standards such as the website development language HTML5 which aims at responding to today's and tomorrow's websites growing requirements. The last version was released in 1997 even if some updates have been made since then. According to its developers, HTML5 is more than a simple upgrade and its high level of compatibility across systems and browsers is the evidence that researchers and developers are currently drawing the path to this expected harmonization. This previous and current lack of harmonization and thus of multiplatform portability is the essence of the coming challenges and of this thesis.

2.1. The Digital Ecosystem: current and upcoming situation

This section describes how the digital ecosystem is complex. The following list of questions is a relevant starting point (NIXON, 2011). From a websites developer's point of view: do I use a PC or a Mac? And which browser do I use? On the user side: which device do they use to access the Internet? Several for sure. And are they using the same browser, screen, and operating system as I do? This is highly unlikely that they all do. Therefore, developing, adapting and maintaining a website in this multiplatform environment has never been more complicated. To achieve this, having different and specialized support teams are almost inevitable. This issue induces unarguably financial issues that have negative repercussions on websites' multiplatform portability and users' satisfaction.

2.1.1. Multi-devices landscape and specific operating systems

In itself, the figure 15 (see appendix 1) represents the entire situation's complexity. It has been conceived from data generated on an 'average' Monday in the United Kingdom (13th February 2013). This represents that during a day, a single person can access its favorite website through different platforms with different screen size and running specific operating systems and browsers. Even thought the use of all devices increase during the day to reach their peak at night, there are significant discrepancies between them according the considered period of the day. Indeed, smartphones surpasses others devices during the commute period while tablets represent the highest peak of the day at night, surpassing others by far. With no surprises, PCs dominate working hours.

However, indentifying periods in which a user prefers using such platform or another is not relevant, at least at first. It becomes relevant when you consider others dimensions such as the environment and the user (e.g. integration of special content or games for commutes). These additional dimensions will be considered in the next chapter.

The first crucial question is, before considering other aspects: does the website or the application can be run properly on every platform and browsers? Indeed, as shown in the next section 'What users expect from these technologies?' the majority of users are already and increasingly expecting a comparable experience on every platform they use. Criteria are obvious and easily understandable: easy access, loading time and appropriate formatting, for the most relevant ones. If problems occur, it induces undesirable users' behaviors towards the concerned website or application. At first, they will abandon trying to access the website, directly or after few attempts. Subsequently, it is also unlikely they will return and will even less recommend the specific website or application.

Description, trends, relative figures as well as absolute figures will be provided for each device. Moreover, emerging and upcoming mobile devices confirm that current achievements are not an end point. Nevertheless, an exhaustive list of every devices brands and versions would be useless in this context but what differentiates them the most is the operating system they are running. In the field of web development, preferring such operating system or another is a matter of software, tools and hardware that developers use to develop a website. On the users' side, different operating systems become indistinguishable since more and more is done through browsers that are today available on every device.

This section provides relative figures as well as absolute figures. Indeed, reparation within a specific component can vary while the use of this component is either increasing or decreasing.

2.1.1.1. Desktop and laptop computers

Table 1 (see appendix 2) summarizes PCs and laptops different types and the operating systems they are running. Considering them, it is useless to differentiate stationary computers and laptops computers since these are just stationary computers without electrical wires.

Windows and Mac OS X represent more than 90% of the running operating systems in 2013. In comparison with less used operating systems such as Linux, Windows has had the advantage to be omnipresent for decades and Apple produces its own devices (e.g. Mac, iPhones, iPad) and their own operating systems at the same time. Operating systems based on Linux have the particularity to be developed according to a free and open source model. They are maintained by a community gathering volunteers worldwide.

However, mobile devices' tremendous uptake is drastically changing trends previously established. Chart 1 represents the relative evolution in the use of these operating systems over five years (W3SCHOOLS, 2013). Platforms counting for less than 0.5% are not represented.

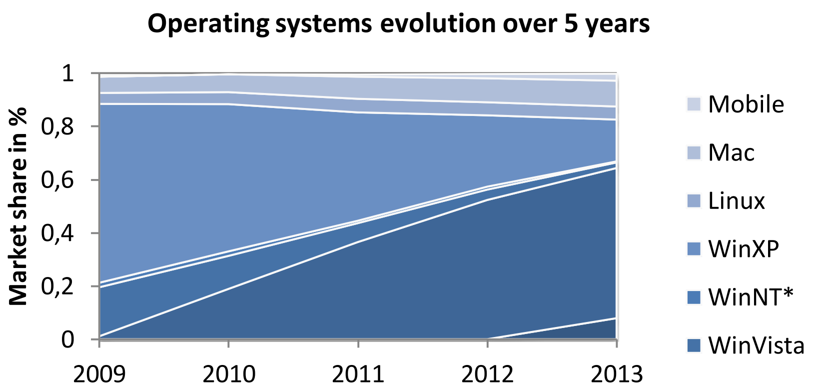


Chart 1 - Operating systems evolution over 5 years (W3SCHOOLS.COM, 2013)

* NT includes operating systems such as Windows Server 2003 and 2008 as well as Windows 2000

As a matter of facts, Windows owns the lion's share (82.5% in 2013), even if slightly decreasing and new versions replacing logically older ones as they go along. Mac's and

Linux’s shares appear stable over the period and even faintly increase.

Over a longer period of time, conclusions are the same: Windows is dominating the market even if decreasing (93.7% in 2003 against 82.5 in 2013) to the advantage of emerging operating systems: Linux (2.6% in 2003 against 4.9% in 2013), Mac OS X (2.2% in 2003 against 9.7% in 2013) and most of all the mobile operating systems.

As shown in the next section, mobile operating systems proportion has already reached 2.6% after few years, showing the success of mobile devices. Almost doubling each year, this uptake is remarkable if compared to Mac’s and Linux’s evolution in progress for decades now.

In terms of absolute figures, "The battle for consumer wallet share continues between different devices. The PC (Mac included) is the first to fall by the wayside as usage patterns shift toward smartphones and tablets. This ongoing trend will have a profound impact on the size of the installed base of PCs" (GARTNER, 2013). Figures are explicit: PCs shipments in Europe decreased by 20.5% in the first quarter of 2013 in comparison with the first quarter of 2012, from 15,483 thousands to 12,307 thousand. This decrease concerns both the professional PC market and the consumer PC market, falling respectively by 17.2% and 23.7%, as well as desktop PCs and laptop PCs, falling respectively by 13.8% and 24.6%.

The financial crisis unarguably plays a role but Corporate PC refresh cycles are not predictable anymore and have been delayed because IT managers consider other technologies such as mobile devices, cloud computing and Windows 8, optimized for the use of touch screens, represent significant expenses for a company or a consumer while “many businesses are still migrating from Windows XP to Windows 7, due to the April 2014 deadline when Microsoft ends support for Windows XP” (SARAN, 2013). On the consumers’ side, they tend to buy tablets instead of replacing or upgrading their older PCs. However, even if this shift has been in progress for several years, these figures have to be considered carefully. Some companies will always need powerful computers and with the emergence of new economies, desktop and laptop computers will still be used in the future, as depicted by chart 2.

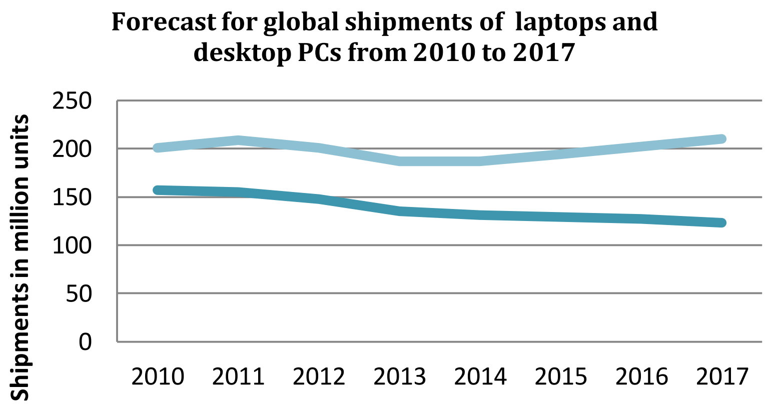


Chart 2 - Shipments of laptops and desktops PCs from 2010 to 2017 (IDC, 2013)

2.1.1.2. Smartphones & Tablets

Mobile operating systems, running on smartphones and tablets, focus mainly on the wireless connectivity management and the interface management.

a) *Smartphones*

The operating systems landscape was wider at first as many companies had developed their own mobile operating systems. Nevertheless, many of them were abandoned, under the supremacy of Google, Microsoft and Apple.

Table 2 (see appendix 2) represents Smartphone Operating systems' shares evolution between the first quarter of 2013 and the corresponding period of 2012. The landscape has significantly changed within one year. As a result of Nokia's switch to Windows Phone, Symbian lost its biggest vendor and its shares dropped sharply from 6.8% to 0.6% within one year. Similar decrease is highlighted for BlackBerry OS but it is not representative as they just sold 1 million devices running their new operating system.

Linux's decrease is justified by some vendors that switched to Android. For the rest, Microsoft Phone, Android and iOS all increased their shipments volume and their market shares except iOS which is losing market shares from 23% to 17.3%. "Although demand remains strong worldwide, the iOS experience has remained largely the same since the first iPhone debuted in 2007" (IDC, 2013). Android remains the strong leader as they own 75% of market shares. Chart 3 provides an overview.

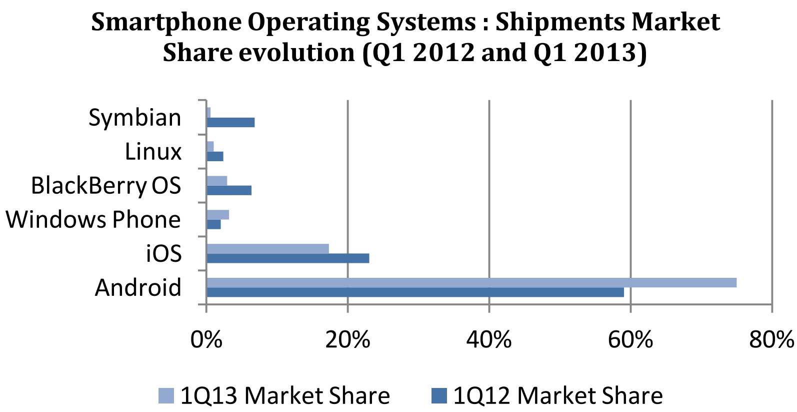


Chart 3 – Smartphones' Operating Systems: Shipment Market Share (IDC.com, 2013)

In volumes, chart 4 shows that Android increased its shipments by 79.5% while Windows Phone represents the most important increase in shipments: 133.33%. Even if they lose market shares, iOS' shipments increased by 6.6%. In total, 216.2 Millions smartphones were sold during the first quarter of 2013, against 152.7 one year earlier. In other words, sales increased by 41.6%.

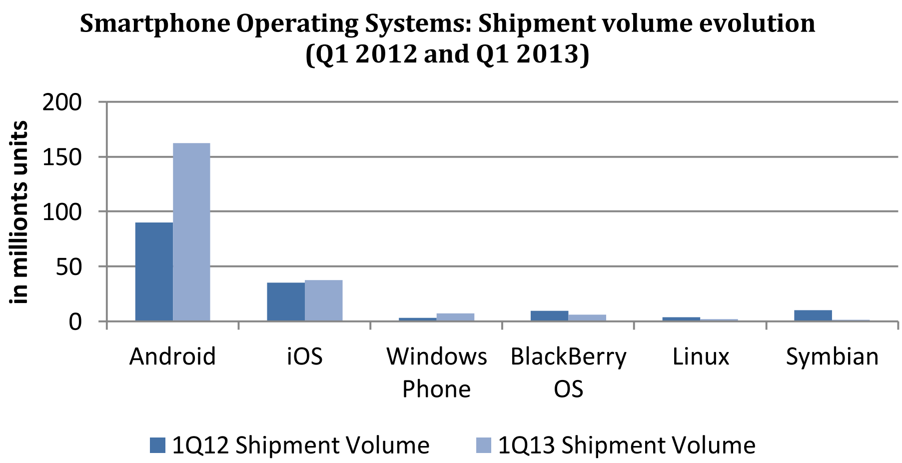


Chart 4 - Smartphones Operating Systems: Shipment Volume (IDC.com, 2013)

b) *Tablets*

Concerning the skyrocketing tablet market, the operating systems' landscape is narrower with only the same major actors: iOS, Android and Windows. An important switch occurred between 2012 and 2013: while Android and Windows were increasing their market shares, respectively from 39.4% to 56.5% and from 1% to 3.3%, iOS market share dropped from 58.1% to 39.6%. Chart 5 sums up this trend (see appendix 3, chart 5).

In volumes, this period has been remarkable as tablets total sales increased 142.4% within one year to reach 48.9 million units. Android's and Windows' sales were the most impressive, with respectively a 247.5% and a 700.0% increase. iOS followed but not at the same rate with a 65.3% increase. More details are available in appendix (see appendix 3, chart 6).

2.1.1.3. The new mix: Computers, smartphones and tablets

"The uptake of mobile technology in our work and private spheres has had a huge impact on the way we perceive and use these technologies. They are no longer just computers on batteries. They have become functional design objects, the look, feel and experience of which we care deeply about, and that we juggle in multitude in our everyday lives" (KJELDSKOV, 2013). However, previous and next figures indicate that personal computers will not disappear in the coming years and that smartphones as well as tablets will spread and even surpass their predecessors, but in a complementary way. "As consumers shift their time away from their PC to tablets and smartphones, they will no longer see their PC as a device that they need to replace on a regular basis" (GARTNER, 2013). Indeed, with a closer look to proportions, there is no complete substitution effect for computers. Population will therefore live with a mix 'computer-smartphone-tablet' in constant evolution, at least for the coming years before new radical evolutions. Chart 7 and table 3 sum up all these coming trends.

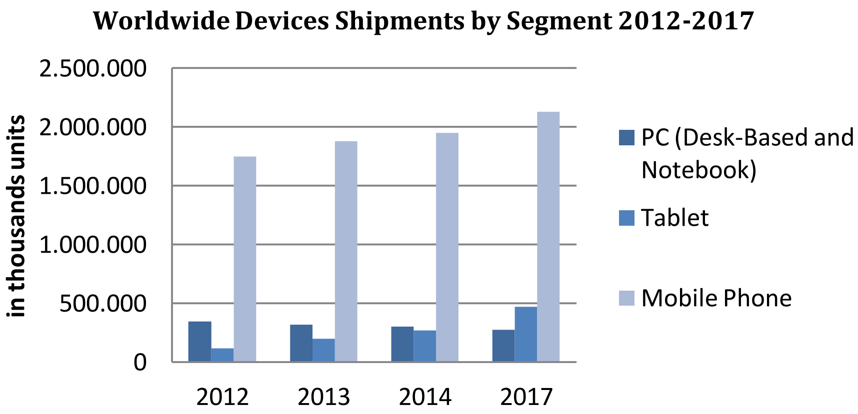


Chart 7 - Worldwide Shipments estimation (GARTNER, 2013)

Worldwide Devices Shipments by Segment (Thousands of Units)					
	2012	2013	2014	2017	2012-2017(%)
PC (Desk-Based and Notebook)	341.263	315.229	302.315	271.612	-20%
Tablet	116.113	197.202	265.731	467.951	303%
Mobile Phone	1.746.176	1.875.774	1.949.722	2.128.871	22%
Total	2.203.552	2.388.205	2.517.768	2.868.434	30%

Table 3 - Worldwide Devices Shipments estimation (GARTNER, 2013)

In 2017, while PCs' shipments will have decreased by 'only' 20%, tablets will have known a 303% increase thanks to the proliferation of cheaper tablets, surpassing computers around 2015 (CHENG, 2013). "Lower prices, form factor variety, cloud update and consumers' addiction to apps will be the key drivers in the tablet market" (GARTNER, 2013). In the meantime, smartphones will have continued its steady evolution with a 22% increase. Thanks to the spread of affordable smartphones, a wider proportion of the population –including emerging markets – will be abandoning their feature phones to replace them with a smartphone. Moreover, smartphones have already crossed the 50 percent threshold in many countries (COMSCORE, 2013) such as the United States, Canada, France (53%), the United Kingdom (64%), Italy (53%), Spain (66%) and Germany (51%). Neighbors and emerging economies will therefore follow soon.

As a consequence, operations systems will also be affected by the shift to mobile devices. Android will keep dominating the market while iOS/Mac and Windows will fight for the second place. Figures are available in appendix (see table 4 in appendix 2 and chart 8 in appendix 3).

2.1.2. Key findings

As a conclusion for this section and for the appendix 8, key findings can expressed as following:

- The shift from PCs (desktops or laptops) to mobile devices is already in progress. However, this is not a complete substitution effect. These technologies are complementary. Smartphones have already surpassed PCs in terms of volume and tablets will achieve the same performance around 2015. Therefore, the Internet which has been built for PCs at first has to undergo radical improvements and adaptations to meet users' needs. In addition to this mobile devices uptake, all the figures also indicate that mobile devices will become soon the major access points to the Internet.

This has induced the development of connected features and components (appendix 8):

- Mobile devices represent such an important sector for users and providers that most PC's browsers coexist with their mobile equivalent. While some have existed for years (Opera, Safari), others have just appeared such as Google Chrome for Android (2012) even if Google dominates the mobile sector.
- As smartphones and tablets have an increasing success, remarkable improvements have been done in the screen resolution and size to offer users a remarkable experience.
- An unprecedented work has been done to spread widely efficient Wireless and data

networks, allowing users to access the Internet whenever and wherever they want. The development of 4G technologies will increase this trend to an unprecedented point. One the one hand, the large adoption of mobile devices will increase mobile connections to such an extent that mobile devices will surpass PC's connections by far. On the other hand, this broad adoption in combination with 4G capabilities allowing easy access to on-demand videos for instance, will multiply mobile data volume.

- To fulfill users' aspirations, the development of wide applications markets has already started and everything indicates this trend will be consolidated in the future.

Therefore, websites' development and conception frameworks have recently changed and two issues arise from this. On the one hand, in a multiplatform environment, developers cannot build websites or applications as they did few years ago or ten years ago. On the other hand, billions of websites have been built since the creation of the Internet and those need to be replaced or adapted. As shown in the next section, this twofold issue is limiting users' experience and introduces the necessity for multidimensional context-aware adaptations.

2.2. What users expect from these technologies?

Previous sections detailed the consequent work that has been made to develop mobile technologies: adapted form factor, mobile operating systems, mobile browsers, mobile data networks and the adapted user interface for the most important ones. However, as depicted by the following figures, older websites is not adapted to be accessed on mobile devices or even worse, some recent websites are not developed to meet these new trends. As a result, users experience is often limited on mobile devices and it induces some undesirable behaviors towards these websites. Subsequently, website owners limit their accessibility and may lose regular visitors, their brand awareness or even worse, market shares. For many years now, a multitude of researches and surveys have been conducted to identify why mobile users are disappointed by web mobile experiences.

This section is structured according to four surveys covering together the period 2008-2012. Therefore, evolutions in users' expectations can be highlighted and they are expected to be stronger as mobile devices' adoption increase year after year.

2.2.1. Loading time and other issues

As a matter of fact, connection speed is currently lower on mobile devices than on stationary computers or laptops. With no adaptations, it leads to a longer loading time on mobile devices. Across all the surveys, it is unarguably highlighted that “the mobile web is disappointing users who have high expectations for mobile web performance and little patience for poor performing sites” (COMPUWARE, 2011). And even if applications – born with mobile devices – are widely spread and continuously developed, “two-thirds say they prefer the mobile web over downloadable mobile apps for accessing Consumer Products/Shopping and Media & Entertainment content” (ADOBE, 2010) while they prefer “to use mobile apps over mobile websites for map information, social media updates, e-mail, and banking information” (KEYNOTE, 2012). Therefore, applications development cannot be done to the detriment of websites’ quality. Further, “most mobile users expect to make sacrifices in terms of content depth and feature-richness in exchange for the convenience of anytime, anyplace mobile Web access. One thing mobile users will not sacrifice, however, is speed” (COMPUWARE, 2011).

2.2.1.1. Loading time expectations

Over time, an increasing proportion of users expect the loading to be almost as quickly on their phone as on their PCs: from 58% in 2009 to 71% in 2011 (see chart 15).

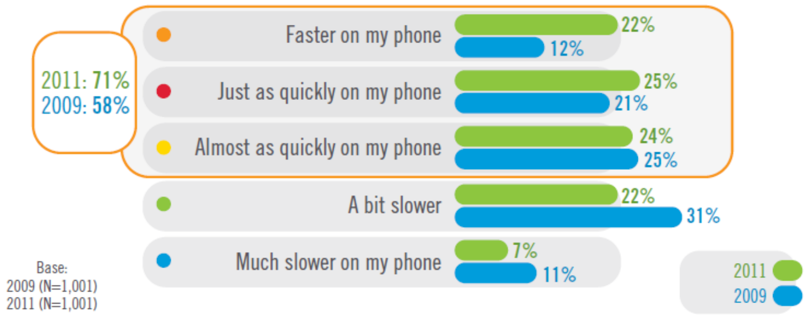


Chart 15 – Relative expectations for loading time on mobile phones (COMPUWARE, 2011)

A more recent survey indicates that this proportion still increases and that the loading time is the number one frustration for mobile users (Keynote, 2012). Figures can be puts on this trend, if loading time is expressed in seconds:

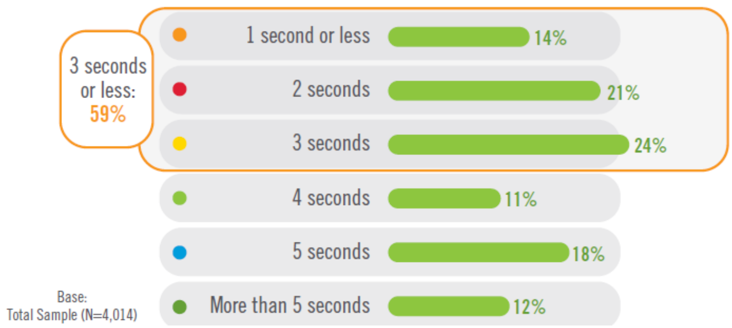


Chart 16 –Loading time expectations on mobile phones (COMPUWARE, 2011)

Chart 16 highlights that 59% of users expect the loading time to be 3 seconds or lower, 29%

are more patient and can wait up to 5 seconds and only 12% can be considered as indifferent regarding the loading time. This trend is confirmed by another survey highlighting that 64% of users "want a site to load in less than 4 seconds. Furthermore, users are even more demanding for websites and applications running on tablets: 6 of 10 tablet users want a sub-three second download" (KEYNOTE, 2012). Hopefully, users often wait 1 or 2 additional seconds in comparison with their expectations but in general, this additional time is not enough as "average website takes more than twice that amount, at 9 seconds" (JOHANSSON, 2013).

Further, surveys indicate that users would use their mobile devices more often if the experience was improved. However, 77% of big companies' mobile sites have loading time longer than 5 seconds (COMPUWARE, 2011).

2.2.1.2. Other access issues

Alongside the loading time, users often face other problems while accessing websites or applications on their mobile devices. As depicted in chart 17, many problems surprisingly happen more on applications than websites: frozen page, crashed page, error, page not available or does not function as expected. Hopefully, applications seem to load quicker than websites. Finally, formatting issues occur only on websites as applications are initially designed for mobile devices.

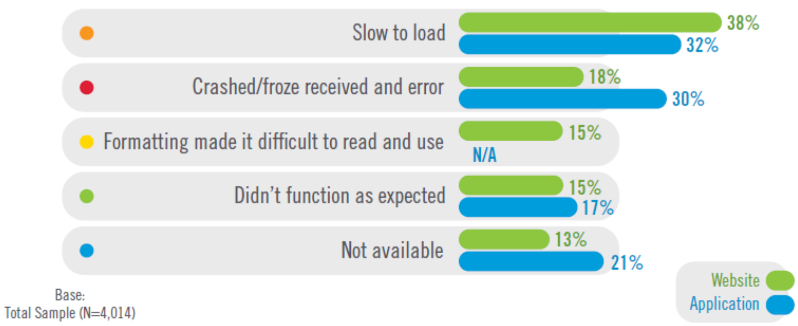


Chart 17 - Access issues on mobile phones (COMPUWARE, 2011)

2.2.2. Disappointed users' behaviors

While users are facing previously mentioned access issues, undesirable behaviors take place with obvious consequences on the preset strategy: loss of regular visitors, loss of brand awareness or even worse, losses of significant market shares and revenues.

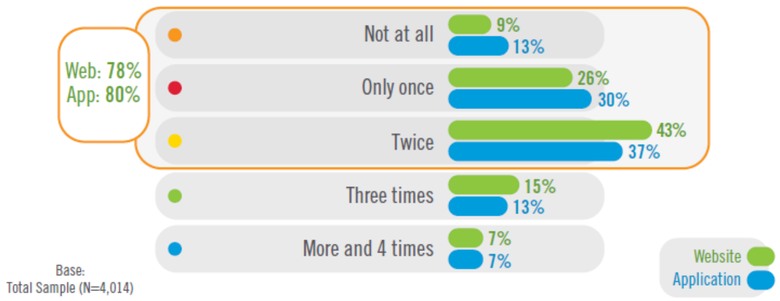


Chart 18 - Retry rates (COMPUWARE, 2011)

Users are demanding and eager. As a result, if they face any issue, they will attempt accessing a website or an application only few times before leaving. As depicted by chart 18, 80% leave applications and 78% leave websites after two attempts. However, figures show that users are even more demanding towards applications than websites: 43% leave applications after one or no attempt against 'only' 35% for websites. As a nightmare for companies, "40% said they would likely visit a competitor's mobile website instead" (EQUATION RESEARCH, 2009).

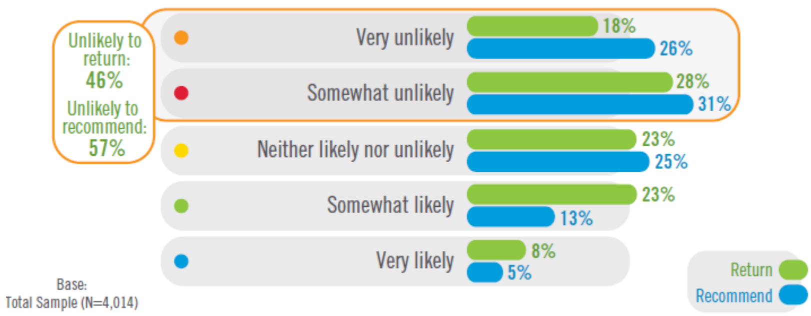


Chart 19 - Return and recommendations rates (COMPUWARE, 2011)

In 2011, 46% of users were unlikely to return and were even more unlikely to recommend the specific websites (57%) if they experienced problems during the last visit. Surprisingly, 31% of users are likely to return and 18% are likely to recommend a website or an application even if they faced issues (see chart 19).

2.2.3. Key Findings and recommendations

Firstly, users are demanding and eager. This means they generally expect the loading time to be at least similar on mobile devices than on stationary devices. If the loading time is too long or another issue occurs (e.g. frozen page, formatting issues) users will attempt reloading only few times before exiting. In the worst case, they will access competitors' websites. Secondly, bad experiences on websites or applications may have a multiplication effect as disappointed users are not likely to return and even less likely to recommend the specific website or application.

From a strict financial and economic point of view, recommendations can be clearly made in order to optimize the global mobile web experience and therefore maximize the ‘mobile opportunity’. However, these recommendations remain valid for every sector using the Internet as a tool to attract and retain user.

It is unarguable that a website or an application which does not work as expected or perform slowly will fail at retaining customers. Expected speeds are those of mobile behaviors such as waiting for a bus, in a line or in front of a ‘Do not walk’ sign. In no case a bad mobile experience will impact positively generated revenues. Moreover, it increases support costs and damages the brand image.

As a starting point, three questions must be addressed from your customers’ perspective:

- Does your website perform according to users’ expectations?
- Are your competitors providing available and satisfying web services?
- Does the user experience reach the same level on every platform your customers use?

These questions become imperative as the adoption of mobile devices quickens worldwide. Companies should investigate and invest in new users’ behaviors towards mobile technologies. Firstly, previous investments in desktop-based devices are not lost. On the contrary, knowledge accumulated can be used to approach wisely information systems and websites based on mobile devices. Secondly, in order to maximize users’ experience, a holistic point of view must be adopted to consider all the platforms and investments are required in order to optimize the mobile experience. Finally, the most important process to integrate is the monitoring of users’ expectations and satisfaction levels. As mobile technologies are still in its nascent days, behaviors and trends are expected to change profoundly in the coming years.

As a conclusion, “businesses that embrace the mobile opportunity, offer the most usable features, and provide the fastest, most consistent performance will emerge as mobile leaders in their category” (COMPUWARE, 2011).

2.3. First steps in the harmonization process: the World Wide Web Consortium

While new mobiles technologies are currently spreading worldwide, previous section highlighted that the lack of technical harmonization and the lack of multiplatform adaptations provide end-users with disappointing user experiences.

In order to tackle these issues, the World Wide Web Consortium (W3C) was created in 1994 within the Massachusetts Institute of Technology (MIT). The Serenoa project is part of W3C. They try to enforce compatibility standards and general agreements within the Internet industries. In 2012, W3C had 379 members with among them, multinational companies and large organizations such as Apple, eBay, Facebook, Google, Microsoft Corporations, NASA, Nokia, Samsung, CERN, Twitter and Yahoo (W3C, 2013). Concerning websites standards, different versions of HTML and related technologies have been provided by many vendors, leading to inconsistency in the way websites are displayed.

As a result, in 2007, the development of the HTML5 recommendation started, aiming at supporting latest media (e.g. geolocation, new APIs, offline web applications) cross-platform compatibility and aiming at becoming the new web development standard. Both W3C and WHATWG (Web Hypertext Application Technology Working Group) are cooperating on this project with different approaches. In 2008, segments of HTML5 were already implemented in several browsers. Many features have been designed by taking into consideration new mobile technologies and their weaker capabilities. By September 2011, 34 of the top 100 Websites worldwide were already developing their websites with HTML5. In 2012, it becomes a candidate recommendation (second step after the draft version). From this point, W3C focuses on HTML5 with a slower approach which aims at providing a definitive standard while WHATWG continues developing HTML5 as a living standard. It means that continuous updates will be implemented and that new features can be continuously added. Nevertheless, it means that features cannot be removed. According to its developers, HTML5 is more than a simple upgrade. Combining HTML5, CSS3.0 and JavaScript improvements, this recommendation is a multi-device, multi-platform and multi-browser strategy and provides a high level of compatibility across systems and browsers. This technology can be summed up as “Develop Once, Deploy Everywhere” (W3C, 2013). They plan to propose the full specification (third step) of the HTML5 recommendation in 2014.

Its adoption is increasing worldwide according to a recent survey covering 4.043 web developers (KENDO UI/ TELERIK, 2012). Figure 15 displays the key findings.

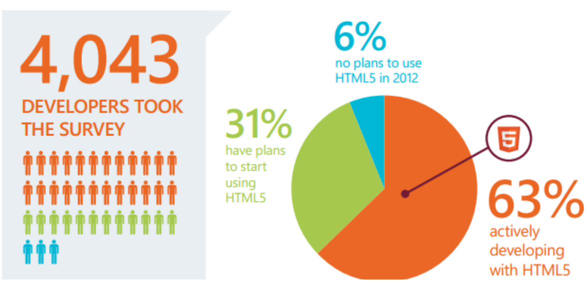


Figure 15 - HTML5's current adoption (Kendo UI/Telerik, 2012)

In 2012, 63% of surveyed developers were already and actively developing web applications and websites with HTML5 while 31% were planning to start using HTML5 by the end of 2012. This is empowered by the fact that HTML5-enabled phones (fully or partially) are expected to reach 1 billion in volume in 2013 (WHITNEY, 2011). Moreover, 82% of developers think HTML5 will become important for their job within 12 months. “Even the 6% of the developers who are not actively using HTML5, find it important in the next 12-24 months” (KENDO UI/ TELERIK, 2012). The Familiarity of languages, the cross-platform supportability and performances of HTML5 are the three main reasons of adoption, alongside with slower costs of development and other features (see chart 20 in appendix 3).

In the browsers’ side, major browsers have nowadays integrated most HTML5 features in their most recent versions (see table 7 in appendix 2). Nevertheless, integration is still in progress for some of them (CANIUSE.COM, 2013). Chart 21 points out that some features are widely implemented while others are partially or not yet implemented.

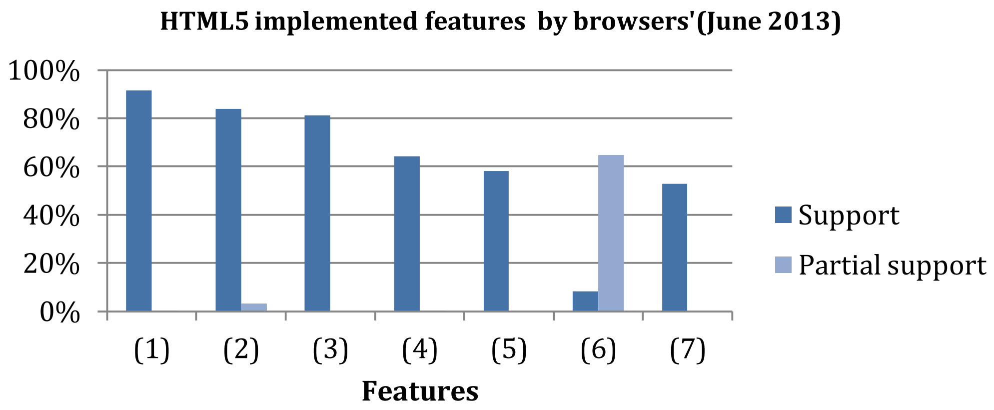


Chart 21 - HTML5 implemented features by browser (CANIUSE.COM, 2013)

(1) Web Storage, (2) New semantic elements, (3) Inline SVG, (4) Server-sent DOM events, (5) WebM/VP8 video format, (6) HTML5 form features, (7) CSS Canvas Drawings.

From both sides – web developers and browsers – , figures indicates that the harmonization towards HTML5 is in progress.

CHAPTER III: multidimensional adaptations and challenges

Previous chapters have a strong focus on the platform dimension. However, this chapter will provide insights in additional dimensions induced by the proliferation of mobile devices: their users and their surrounding environments.

At first, the role of the context and its impact on previous and current researches will be highlighted. Subsequently, this chapter will introduce definitions of context and multidimensional adaptations provided by the literature as well as the related challenges of such a multidimensional approach. Finally, this chapter provides the three adaptation dimensions to be addressed.

Thanks to this approach, this chapter introduces the second part of this thesis. In combination with developed cross-platform tool such as HTML5, these researches and definitions represent the starting point in order to address multidimensional context-aware adaptations. Indeed, in the second part of this thesis, some selected adaptations techniques will be described and implemented for each dimension.

3.1. The role of the context induced by mobile devices

For decades now, it has always been meaningful for developers in mobile computing, or more broadly in human-computer interaction that systems are able to be aware and take into account context's characteristics. Regarding these contexts of use, the mobile and complex nature of mobile devices is a real challenge in comparison with previous stationary devices. These radical changes induced by mobile devices and their multiple contexts of use have influenced other related disciplines within their theory, their technology and their methodology: in mobile computing, in systems' development and design, in usability evaluation and in user experience research for instance (KJELDSKOV, 2013). This influence has produced different results within these different fields of research.

In mobile computing, the main challenge has been to identify and describe the different contexts of use, at least theoretically. Then it has been to study empirically and analyze in which ways contexts of use are relevant and generate a full understanding (DOURISH 2004, DEY 2001, MCCULLOUGH 2004, KOSTAKOS et al. 2009).

In the field of systems' development and design, taking into account the context of use has led to several challenges such as the creation of context-aware systems and context-aware methods and all the related theoretical and technological support.

When it comes to evaluate usability in mobile computing, the challenge has been to turn empirical researches into well-adapted usability tests conducted in actual settings. Obviously, the emergence of mobile devices has modified techniques and methods previously used.

Finally, mobile computing challenged the field of user experience research to understand how different and dynamic contexts of use impact users' experience towards this plethora of mobile devices. Subsequently, the challenge has also been how to improve constantly the user experience.

This advance in several fields of research indicates that contexts of use definitions are crucial as well as their right implementation in practice that influence deeply the way users experience new technologies. Nowadays, population can witness significant advances that have been made in both theoretical and practical fields of mobile computing research but in practice, the next step is to create an holistic context-aware user experience, responding to users' expectations and fitting their context of use while remaining still pleasant as well as useful. Unfortunately, the majority of books available on this topic are desktop-computers-oriented and there is no such equivalent literature yet including other mobile devices in the equation of a digital ecosystem. Literatures do exist but are limited to the focus on specific devices and becomes directly outdated once a new device appears. Case studies also exist but their findings often do not fit the current evolving situation. Finally, some developers also depict single or few specific adaptations techniques on their website without addressing the whole issue.

As a result, information is disorganized and current literature does not provide developers and designers global and consistent basis to either develop their own context-aware adaptations nor recommendations to investigate this field of research. There is therefore a clear mismatch between mobile interaction design theories developed for more than decade now and current practices. The emerging digital ecosystem is not well understood yet and internet applications are consequently created on traditional methods. The main justification is that the current enormous enthusiasm for mobile devices indicates that the sector has not reached a stable point yet and that fundamental researches must keep going on in order to provide strong and stable guidelines and recommendations.

However, in order to address the next step (i.e. the user experience in a constantly evolving digital ecosystem), it is crucial at first to extract the essence of these previous and priceless researches that have been conducted for decades. Secondly, with the help of recent tools which aim at providing cross-platform harmonization and modern capabilities such as HTML5 exposed previously, it becomes possible to build systems taking into account the diversity and the complexity of the current and future digital ecosystem.

This approach sums up the content of following sections. Firstly, next sections of this chapter will introduce definitions of context and multidimensional adaptations provided by the literature as well as the related challenges. Subsequently, the second part of this thesis will identify dimensions to be addressed as well as corresponding adaptations techniques.

3.2. Context definitions: multidimensional adaptations

During decades, researches have provided many definitions of context but debates are still in progress. Table 20 sums up the essence of these definitions. They are detailed in appendix 10.

Context's dimensions	**Dimensions' features**	**Source**
Environment	Location	SCHILIT (1994)
Any Entities relevant in the interaction (Person, Place, Object)	User Applications	DEY (2001)
Physical factors	location (absolute relative position), infrastructure (computational resources), and physical conditions (noise, light)	SCHMIDT et al. (1999)
Human factors	Users (profile, mood), social environment (group dynamics), users' tasks (current activity, goal)	
Environment	Pathways and landmarks	MCCULLOUGH (2004)
Cannot be stable.	It depends on users' activities and must be continuously updated	DOURISH (2004).

Table 20 – Context of use's definitions

At that time, the most elaborated context of use' definition is the one provided by SCHMIDT et al. (1999). Indeed, as depicted by their book, context is more than location. This definition will be used in the next chapter to identify dimensions and adaptation techniques.

Furthermore, by dividing user interaction with systems into ‘Seven stages of Action’, NORMAN (1988b) allows developers to decompose users’ interaction process in different smaller steps (see appendix 10). Finally, despite its age, Dieterich’s taxonomy is still valid for analyzing users’ behavior towards interface adaptation techniques (LOPEZ-JAQUERO et al, 2008). It states that four steps are needed to achieve every adaptation: the initiative, the proposal, the decision and finally the execution. Details are provided in appendix 10.

Here below are some other definitions:

- “Adaptive system adapts automatically its behavior to the user” (LORENZ et al. 2000)
- “Systems which reflect some features of the user in the user model and apply this model to adapt various aspects of the system to the user” (BRUSILOVSKY, 1996).
- “Adaptive hypermedia applications are complex software systems, whose development process demands an exhaustive feasibility study, adequate planning and experience in the construction of hypermedia applications, user modeling and adaptation techniques” (KOCH, 2001).
- “Adaptability means the capacity of a system to behave according to the context, users' needs and preferences. This criterion is divided into: Flexibility and User Experience” (BASTIEN & SCAPIN, 1993).
- "In the physical world personalization can be very difficult. Roads, buildings and similar systems cannot easily adapt to individuals needs after they are built. But electronic systems ‘can’ if we can just start to see them as serving our needs and not the other way around. The ultimate goal of technology use has to be to make life better, to enable us to do things we could not otherwise do" (HEALTH, 2009).

3.3. Context and multidimensional adaptations challenges

Definitions given above provide limitations as they do not provide guidelines to implement dimensional adaptations. As a result, context has always challenged researchers to extract and capture its dimensions as well as transform them into models. Then, context has challenged them to interpret these models and make them useful for the implementation of context-aware and responsive information systems.

Utopian information systems are seen in a very simple but demanding way: “the idea of an adaptive interface is straightforward. Simply, it means the interface should adapt to the user rather than users adapt themselves to the system” (NORCIO & STANLEY, 1989).

However, one of the most important challenges is to automatically detect user preferences and adapt user interfaces accordingly. Moreover, as everyone has different behavior and personality, it can produce enormous amount of data. Indeed, “some of these parameters such as users’ preferences (e.g. font type, background color) depend on the specific user while others such as users’ context or actions, do not. Moreover, all these parameters vary over time which makes them even more difficult to manage” (MITROVIC, ROYO, MENA, 2005). This can be solved by using users’ categorization.

Another related challenge is the gap between current mobile devices capabilities (e.g. processing speed, storage space) and users’ expectations for adapted content and quick downloading time. In that context, real-time adaptations to a wide range of environmental factors – which change rapidly, if not continuously – can lead to system trashing. Therefore, adaptations’ performance and utility have to be significant in order to justify such real-time and demanding adaptations.

Finally, some type of system (e.g. anti-virus or anti-hacking systems) can neither predict the form nor the content of future attacks. In this context, some systems cannot be fully adaptive by nature.

This part gathers all the information required to enter the practical part, putting all the previous concepts, definitions and statistics into practice.

Second part: adaptation techniques: implementation and discussion

The first part provides deep insights in the mobile computing history, in the current digital ecosystem, in users' behaviors towards these emerging mobile technologies and in researches conducted for decades in the field of context-aware adaptations. Now everything has been taken into consideration, the second part gives insights into the practical implementation of multidimensional adaptations techniques.

Dimensions to be addressed and their specific adaptations techniques have to be identified. On the one hand, the previous chapter has introduced useful definitions of context and related challenges depicted by decades of researches. From these, context's dimensions can be extracted: the platform, the user and the environment. On the other hand, the second chapter provides key findings on users' experience. They expect at least an equivalent user experience while browsing on mobile devices than on stationary devices. Taking mobile devices' capabilities and features into account, it induces reducing the processing time in order to cut pages' downloading time as well as reformatting web applications and their interaction design according to the device used. In other words, users complain at first on platform-oriented issues (i.e. downloading time and formatting) before any other thing. A focus will therefore be set on this dimension.

In the chapter 4, the three context's dimensions will be identified and detailed: the platform, the user, the environment and corresponding questions and challenges will be mentioned. A specific order of implementation for these adaptation techniques is also suggested. Afterwards, in the chapter 5, the following methodology will be applied. At first, adaptations techniques will be chosen from the Serenoa project's Working Area (the white paper is available in appendix 9). This project gathers more than 150 adaptation techniques and a selection is therefore necessary according to their relevance and the skills required. Around 25 techniques have been selected for their wide scope of application, their high level of relevance in many types of applications as well as the skills required to implement them. Some have not been fully or partially implemented and motivations for not implementing them are provided in the table 21 (e.g. time consuming, skills required). They have been either analyzed on different cases or implemented on the Serenoa's study case: the car rental website (see appendix 7). They all have been evaluated. HTML, CSS and javascript files are available on the CD attached to this thesis. Depending on the case, implemented techniques are illustrated with previous researches, pieces of codes and screenshots and evaluated by providing pros and cons for each of them.

Finally, in the conclusion, selected adaptations techniques are gathered in a feature diagram offering users and developers a global overview of multidimensional adaptations techniques ordered in the suggested order and their conditions to be applied for both developers and users. Last sections provide advantages and shortcomings of such approach and open doors for further researches.

Chapter IV: Context's dimensions identification

Previous context of use's definition by SCHMIDT et al. (1999) and context's dimensions identified in the Serenoa project working area lead to the analysis of three context's dimensions for this thesis (see figure 16). Accordingly, selected adaptations techniques for each dimension (see table 21) respond to following interrogations and challenges. The suggested order of implementation is afterwards justified.

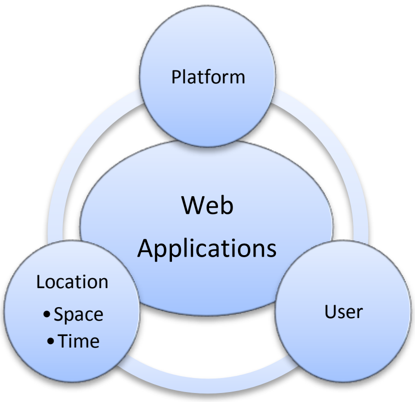


Figure 16 – Web applications and context's dimensions

1) The platform (which device?) has to be addressed at first to meet previous users' recommendations. The challenge for interfaces is to be adaptable across platforms to such an extent that the release of a new type of device would lead to no or few modifications. Three aspects have to be studied for every device: the layout, interaction designs and the loading time optimization.

2) The absolute location (where?) and the relative location (who/what is present in the surroundings?). The challenge is to find the right balance between the flow of data required for a specific adaptation and devices' capabilities. Real-time adaptations would generate too much data for current technologies. The time dimension lies within the location dimension as the time depends on when you are on earth. Indeed, everything moves in a space-time continuum and taking these two dimensions separately would make no sense. "At 3pm" and "at my place" only make sense together. Without this space-time continuum, GPS would not be able to compute drivers' location.

3) The user as a final feat (who? which traits of characters? And what is the current activity?): "it means that the interface should adapt to the user rather than the user adapting to the system" (NORCIO & STANLEY, 1989). The challenge is the users' uniqueness: while some features are constant (e.g. gender), others are specific or may change (e.g. age, font types' or font sizes' preferences, personality) and others change constantly (e.g. current activity, mood). Once again, these continuously changing features would produce an enormous amount of data, surpassing current devices' capabilities. Nonetheless, the current challenge is how to capture these varying features (e.g. how to capture user's mood or personality?). This can be solved through users' categorization.

However, from a developer's point of view, these dimensions should be implemented in the previously suggested order, from the most general adaptations to the most specific ones. It does not mean developers have to implement all of them but this order allows continuous improvements in a logical order. Even if the process does not have to be totally serial, this order is justified by the fact that some adaptations are universally accepted and understandable (e.g. adapting design to the screen width) while other adaptations require previous or cross-dimensions adaptations (e.g. users' current activity cannot be extracted if geolocation is not implemented) as well as users' agreement (e.g. location-based advertisements). From another perspective, it would not make sense to develop a web application providing advanced location' and user's adaptations if users are not even able to browse properly in an adapted and user-friendly interaction design on every device.

In that context, cross-platform adaptations have to be implemented at first as a solid basis for further work. Then, users' adaptations come after location's adaptations as users' characteristics and activities are way more specific than a specified location. Indeed, it is much easier to adapt web applications' content to a specific location without taking into account users' preferences, showing affinities to specific locations and contents.

Therefore, a more adapted representation is required and is more representative of the global context (see figure 17). Even though these layers have to be implemented in a specific order, each has to take into consideration next layers and has to be adapted accordingly (e.g. variable values instead of fixed ones). This order also helps to structure this chapter. Of course, the final product may focus on specific dimensions depending on the website's purpose. Everything will be clarified and developed for each dimension and its specific adaptations' techniques in the next chapter. These techniques have been chosen on the Serenoa project's working area and implemented in the car rental website or analyzed in different cases. It will be illustrated with pieces of code and screenshots or previous researches. Afterwards, evaluations and recommendations are provided for each technique.

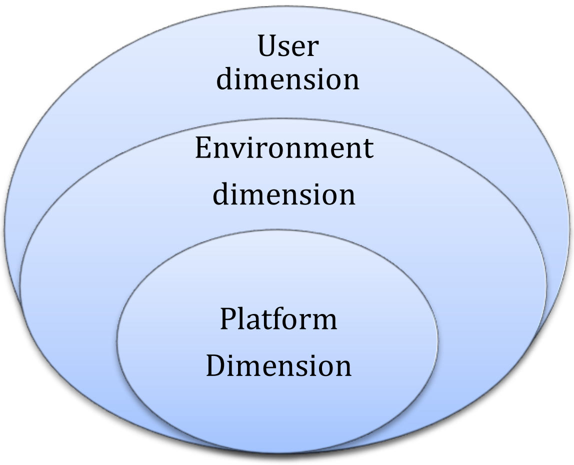


Figure 17 – Implementation structure from the less specific dimension to the most specific one

Chapter V: Adaptations techniques for each dimension

Now the theoretical and statistical materials have been gathered, next contents detail conceptualizations and adaptations for each dimension. Table 21 sums up context's dimensions studied, corresponding selected adaptations techniques and to which extent techniques have been considered. They are ordered following the suggested order of implementation (see figure 30 in the conclusion). However, this order may be different in next sections for practical reasons.

Dimension	**Adaptation steps**	**Technique**	**Level of consideration**
Platform	Loading Time Optimization (0)	Programming best practices	Implemented, tested and evaluated
Platform	Mobile First Approach (1)	Buttons	Implemented, tested and evaluated
Platform		Expandable contents	Implemented, tested and evaluated
Platform		Drag and drop contents	Implemented, tested and evaluated
Platform	Responsive Web Design (2)	Proportional contents' widths	Implemented, tested and evaluated
Platform		Proportional font sizes	Implemented, tested and evaluated
Platform	Media Queries (3)	Setting several thresholds corresponding to different devices	Implemented, tested and evaluated
Platform	Progressive Enhancement (4)	Needs media queries to add contents in function of different devices' capabilities	Analyzed and evaluated

Table 21 – Selected adaptation techniques

Dimension	Adaptation steps	Technique	Level of consideration
Platform	Adaptive Images (5)	javascripts to detect screen's width and send specific images accordingly	Implemented, tested and evaluated
Platform		New HTML5 element: <picture> (Does not exist yet)	Analyzed and evaluated
Environment	Surrounding environments (6)	Brightness regulation	Analyzed and evaluated (embedded into devices)
Environment		Noise recognition	Analyzed (needs deep researches)
Environment	Location-based Services (7)	Location detection	Implemented, tested and evaluated
Environment	Language Detection (8)	IP	Analyzed and evaluated
Environment		HTTP Header	Analyzed and evaluated
Environment		Geolocation API	Analyzed and evaluated
Environment		Ask the user	Implemented, tested and evaluated
Environment		Regional websites	Analyzed and evaluated (time consuming)
User	Current activity (9)	Main activity deduced from users' speed	Analyzed and evaluated (time consuming)
User	Age & Sight's troubles (10)	Font size adaptation	Implemented, tested and evaluated

Table 21 – Selected adaptation techniques

Dimension	Adaptation steps	Technique	Level of consideration
User	Age & Tremor (10)	Tremor adaptation (interaction) Swiping instead of tapping.	Partially implemented (buttons), tested and evaluated
User	Color-blindness (10)	javascripts	Implemented, tested and evaluated
User		Browser's extension	Analyzed, installed, tested and evaluated
User	Blindness (10)	Rethink and reshape applications	Analyzed and evaluated (time consuming)
User	Personality & Mood (11)	Categorization	Analyzed and evaluated (time consuming)

Table 21 – Selected adaptation techniques

5.1. Dimension #1: the platform

Within the platform dimension, two distinct concepts have to be explained at first. On the one hand, the emergence of a plethora of mobile devices allowing two orientations – landscape and portrait – (see figure 18 in appendix 1), lead to the implementation of responsive web contents, also called *Responsive Web Design*. It provides the user with contents that fit its device's screen size. On the other hand, as these mobile devices have lower capabilities (e.g. processor, screen size) than a desktop computer, web applications have to be developed by taking into consideration the lowest-capable device (i.e. mobile phone) and the way people interact with each device: people use a keyboard and a mouse for their desktop and laptop computers, their thumbs on their smartphones and both hands on tablets. This approach is called the *Mobile First Approach*. These two complementary approaches can be merged in a single one: *Mobile First Web Responsive Design* directly applicable to mobile phones.

Subsequently, from this global concept's implementation, the concept of *Progressive Enhancement* can then be implemented to deliver an optimized experience to everyone on every platform. Following sections show that mobile adaptations are not only about screen size.

5.1.1. Insights in existing concepts

5.1.1.1. Responsive Web Design

This concept uses Media Queries which allows the developer to set up dimension breakpoints (e.g. screen width) and corresponding layouts. Each breakpoint (e.g. width in pixels) fit with a range of mobile devices' width but it is not enough for a web application to be fully adaptive as these breakpoints are fixed. Consequently, web applications' design may differ between two devices of the same range. As a result, the developer has to adapt the layout of a website for every device by using adaptive contents such as fluid grids, medias and images that fit the available space within a breakpoint.

5.1.1.2. Mobile First Approach

As mobile devices are both the lowest-capable and the mostly spread devices (see figures in the second chapter), this approach recommends to prioritize the development for mobile devices as a starting point in order to optimize the user experience on every device. It allows developers to focus on prior contents that users really need and therefore allows them to be creative and innovative in the use of new technologies (e.g. touch events) to produce adapted interaction designs. Indeed, if interaction designs are optimized for the use of users' thumbs, it will necessarily be for their mouse and keyboards. Alongside the interaction design, web applications' performances (i.e. processing time) have to be optimized to match lowest capabilities by optimizing HTML codes, scripts and images for instance.

5.1.1.3. Progressive Enhancement

While embracing a desktops-first approach leads to graceful degradation (see figure 19 in appendix 1), progressive enhancement keep everything intact once screen size or the connection speed increase. Moreover, "designing with progressive enhancement involves smartly adding layers of enhancements to a strong foundation in order to deliver an accessible (and hopefully optimized) experience to all" (FROST, 2011). Subsequently, while basic contents will be available from mobile devices, enhanced features can be added brick by brick in function of other devices' capabilities (e.g. high-definition videos). Once the design and the content are adapted to every device, developers can afterwards start thinking about the location dimension as well as the user location.

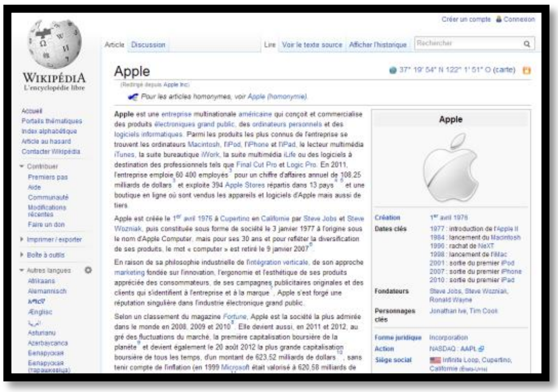
5.1.2. Adaptations techniques

To organize this section, it will be divided into the aesthetical and the technical aspect of the platform dimension. The first part will focus on layout adaptations and interaction design adaptations while the second part will focus on optimizations and adaptations decreasing both the downloading and the processing time for every device. Mobile-first responsive web design is not an add-on to existing websites. It “requires overhauling a site’s foundation and more importantly requires a mental overhaul. This is not a quick fix; this requires careful planning, time, and solid execution. It’s hard. It may sound daunting, but the payoff is huge. Instead of having to create a completely new iteration of a website every time a device gets hot, that time saved can be applied to optimizing the experience for the new context without reinventing the wheel” (FROST, 2011). Hopefully, these concepts are base on web standards (HTML and more recently HTML5) and preexisting web best practices.

5.1.2.1. Adaptive layouts

Nowadays, many Internet users are experiencing the following disappointing user experience while browsing the Internet, with their smartphones for instance. This chapter explains how to manage this issue.

PC
Resolution: 1366 x768



Smartphone (Orientation: portrait)
Resolution: 320x480

a) *Proportional vs fixed layouts*

With fixed layouts, elements do not fit every screen sizes. Here is a relevant illustration with an image:

```
CSS:
img{width:500px;}
```

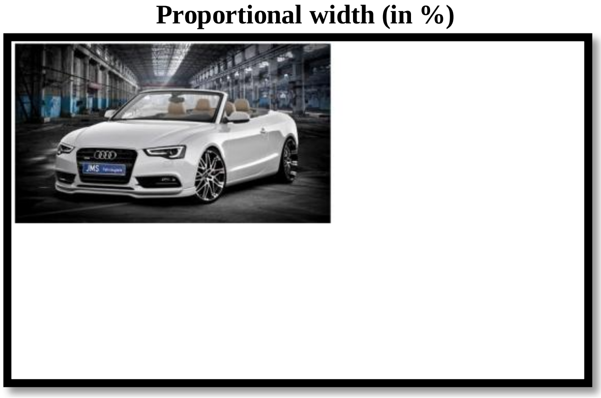
Resolution: 1366 x768
Orientation: Landscape
Fixed width (in pixels)

Resolution: 320x480
Orientation: Portrait
Fixed width (in pixels)

As a cornerstone in the Responsive Web Design, using percentage to express elements' width or other scalable attributes allows elements to fit in every screen size. However, **fixed** limitations can be set to structure the content or to set a minimal width. These adaptations can be applied to every element (e.g. img, video, canvas, border, width, height, padding, margin).

```
CSS:
img{width:90%;max-width:500px;min-width:200px}
```

Resolution: 1366 x768
Orientation: Landscape
Proportional width (in %)



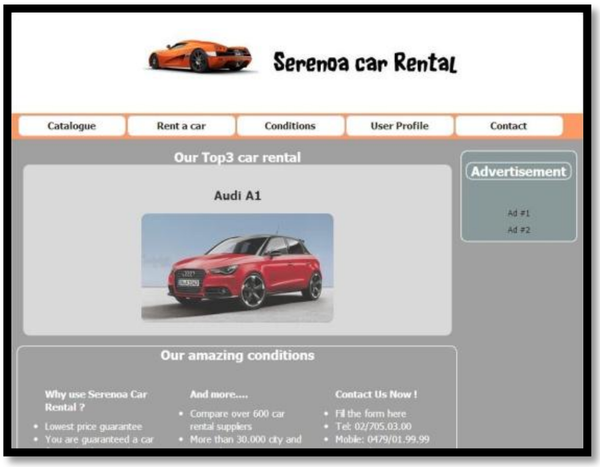
Here, the width is limited to 500px maximum and 200px minimum

Resolution: 320x480
Orientation: Portrait
Proportional width (in %)

However, applied to every element, this adaptation requires repositioning and resizing some elements. But further explanations will be given with the *Media Queries.*

Proportional widths: without any ordering or resizing adaptations

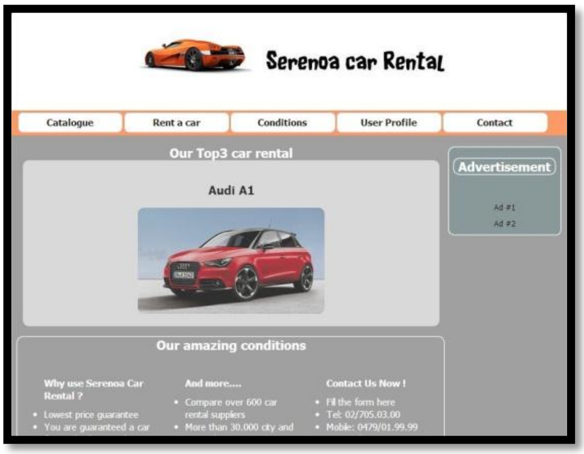
Resolution: 1366 x768
Orientation: Landscape



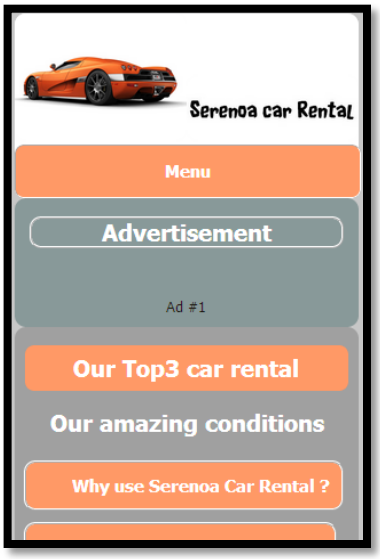
Resolution: 320x480
Orientation: Portrait
Elements need to be resized and repositioned

Proportional widths: with ordering or resizing adaptations through *media queries*

Resolution: 1366 x768
Orientation: Landscape



Resolution: 320x480
Orientation: Portrait



b) *Media Queries*

Before the emergence of mobile devices, elements only required a single value for each feature (e.g. position, width, height). Therefore, a single CSS file was required. Nowadays, elements need several values for each feature, corresponding each to a specific screen width. Indeed, implementing proportional elements fitting every screen is one thing but repositioning and reshaping them accordingly is another one. Several CSS files would be required. These files and their programming language allow developers to set different features' value to a single element according to one or several device's characteristics (e.g. generally the screen width and the orientation) through a CSS technique called: *Media Queries.* Different units can be used to set breakpoints (e.g. generally percentage, inch, centimeter, pixels). One CSS file can be coded for every breakpoint but all the Media Queries can be implemented in the same file, separated by different media queries. Regrouping all the Media Queries in the same file is even recommended by best practices as downloading and processing one single file is faster than several ones (see below: 5.1.2.3. Downloading and processing time optimization).

Table 8 (see appendix 2) introduces criteria used to define breakpoints and also describe the all set of current devices on many features (e.g. screen width, resolution, pixel density). The purpose is to clearly identified breakpoints. Table 8 shows media features and corresponding units on which breakpoints can be set and can be relevant for a general use (W3SHOOLS.COM, 2013). They all accept min-max prefixes.

Other units exist such as '*em* 'which is proportional to the initial value of the font size.

However, Media Queries are only based on initial values and never on results of some computations. Now criteria are known, values for current devices will be helpful to identify which are most relevant criteria to identify breakpoints between devices. From the table in appendix 5 gathering 81 devices' features, some key findings can be extracted. For the last years, breakthrough technologies have blurred the boundaries between desktops, laptops, tablets and smartphones. Indeed if pixels densities are compared to the screen size, no clear distinction can be made between devices and 'dpi' (dots per inch) is therefore not a good breakpoint criterion, at least for the moment. Chart 22 shows that as screen size increases, there is no clear trend for pixel densities.

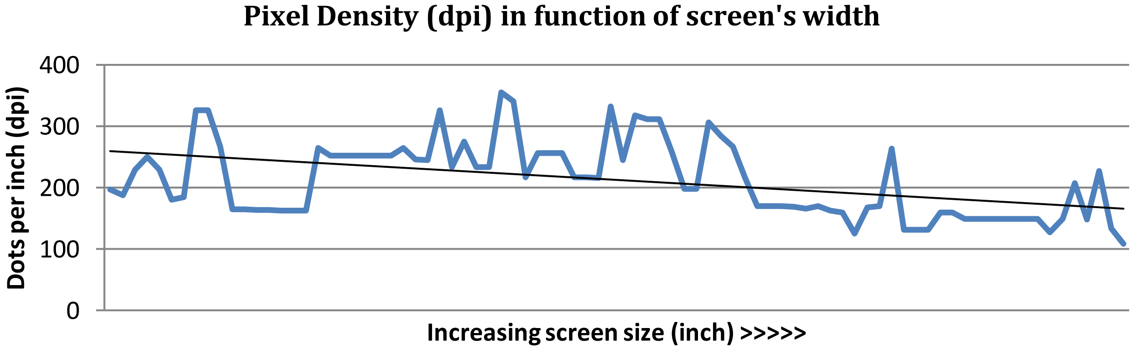


Chart 22 – Pixel densities vs screen size (appendix 5)

The trend is more straightforward concerning viewport width. Indeed, there is a positive correlation between the screen size and the device's width in pixels. However, some peaks appear and using width as a unique breakpoint criterion would lead to some irrelevant situations.

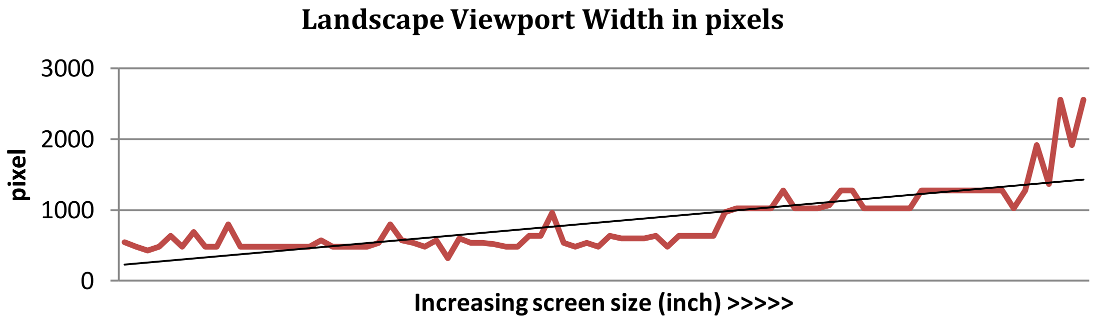


Chart 23 – Screen's width resolution vs screen size (appendix 5)

Which such findings, is that pointless to remember what are the goals of interaction design researches? They aim to provide users with a pleasant user experience, allowing them to interact properly with the contents displayed at a speed matching their expectations. In that context, while 'dpi' features are blurring the boundaries and while width does not give totally clear distinctions between devices; is that nonsense to choose the physical screen size as breakpoint criterion? “Since we are designing for humans, shouldn’t we be thinking about the physical side of human data consumption and designing using this kind of a metric?” (CHELARIU, 2013). Surprisingly, even if it would make sense, developers should not rely on physical measures. Indeed, there are unfortunately two kinds of pixels. On the one hand: ‘Logical pixels’ or ‘Device-Independent pixels’ used in CSS. On the other hand, there are actual pixels varying with “the display resolution and the physical size of the monitor. Therefore, physical inches are not a useful measure, because there is no fixed relation between physical inches and pixels. Instead, elements are measured in logical units” (WINDOWS, 2010). Written more clearly, media queries based on physical measures (i.e. inch or centimeter) would be unreliable as pixel density (dpi) is different on every device. Therefore, two devices with same screen resolution but two different pixel densities may fall under the same media query despite the fact that one of them is twice bigger in size than the other one (see example in figure 20 between the Apple iMac and the Apple Mac Book Pro).

Device	Size (inches)	Resolution	PPI (pixel per inch)
Apple iMac	27	2560x1440	109
Sony Vaio F	16.4	1920x1080	134
Apple MacBook	13	2560x1600	227

Table 9: the resolution criteria leads to errors (appendix 5)

In such context, many developers propose a project-by-project approach while waiting for a more suitable solution. Other developers elaborate their own theory such as the PSINET which stands for The Physical Size Inquiry Non-Exhaustive (CHELARIU, 2013). The formula is:

```
Ratio=min[Width,Heigth]/pixel density
```

According the author, if the ratio is bigger than 5, the device is considered as a large device. If it is close to 5, it is considered as medium-size. In other cases, it is considered as a small device. This is a useful tool leading significant results but developers may have different interpretations of what is a small or a large device. Other thresholds’ values have been extracted from the table in appendix 5 (see chart 26).

The relation between the ratio and the pixel density becomes clearer (see chart 24).

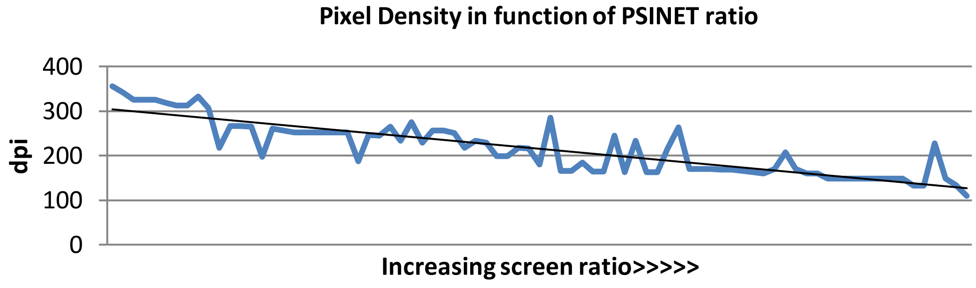


Chart 24 – Pixel densities vs increasing PSNIET ratio (appendix 5)

The relation between the ratio and the width has not fundamentally changed.

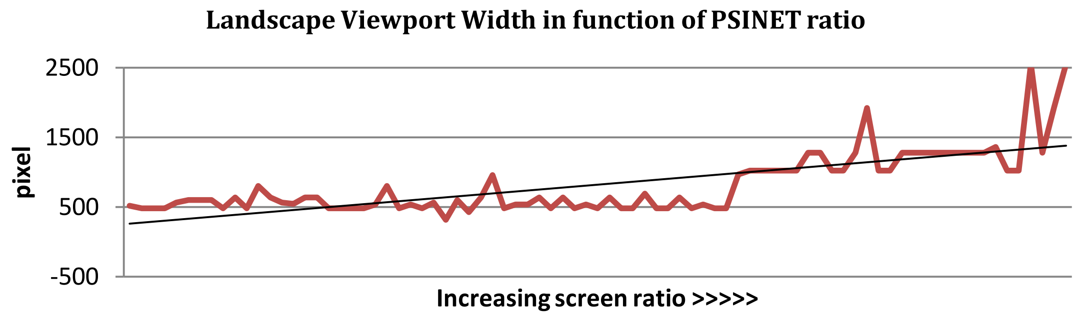


Chart 25 – Screen's width resolution vs increasing PSINET ratio (appendix 5)

If these two sets of data are merged, it produces the chart 26 showing wide range of data in which clear breakpoints' values for both features (i.e. screen width and pixel density) can be identified for the main devices (i.e. smartphones, tablets, and laptops-desktops). Highlighted ratio values correspond to breakpoints between these devices (see appendix 5). Table 10 provides the computed values for these breakpoints that can therefore be use in media queries.

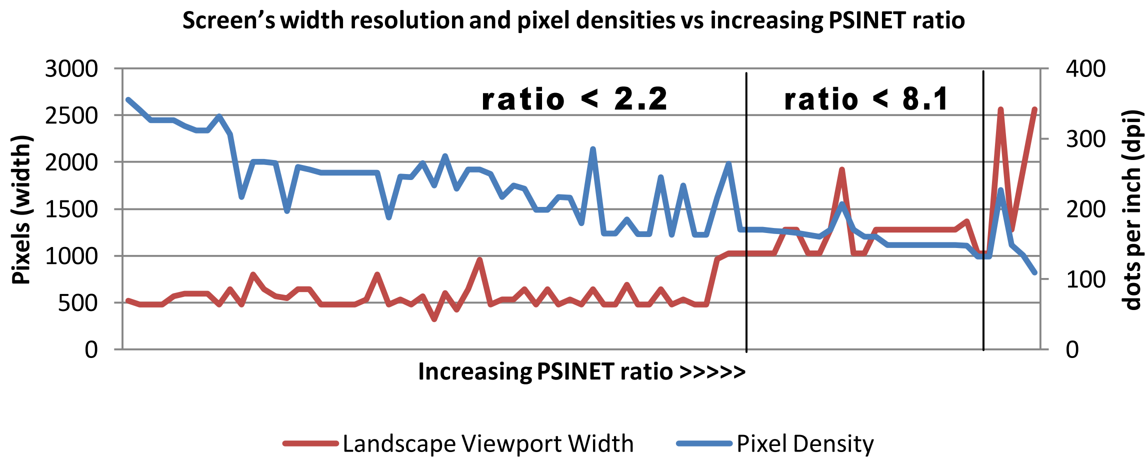


Chart 26 – Screen's width resolution and pixel densities vs increasing PSINET ratio (appendix 5)

Device	Ratio	Min DPI	Max DPI	Min Width	Max Width
Smartphones	ratio < 2,2	163	355	320	966
Tablets	ratio < 8,1	132	264	1024	2560
Laptops/Desktops	ratio $\geq$ 8,1	109	134 (131)	1920	2560

Table 10: breakpoints' values (appendix 5)

As depicted by the chart 26, segmentation is clear for smartphones even if DPI values are overlapping because width values are well-separated from others. However, tablets' values are overlapping on laptops-desktops' values on both features (in red). It is therefore necessary to have a closer look at the table (see appendix 5): iPad and iPad 2 are the cause of this issue (tablets) with a dpi value of 132. Therefore some features may be modified (in green) to make iPad and iPad 2 fit with the tablet category. This kind of manipulation is impossible without significant consequences if media queries use one single feature. For instance, tablets' max-width value should be set at 1919 instead of 2560 to avoid overlapping. As a consequence, several tablets would be considered as laptops or desktops with consequences on interaction designs and user experiences explained in previous chapters. In that case, both features are complementary and allow applying the right CSS rules to the right device. Here are the basic media queries that would be applied according to the criteria.

```
/*SMARTPHONES*/
@media (min-resolution: 163 dpi) and (max-width: 966px) > {}
/*TABLETS*/
@media (max-resolution: 264dpi) and (min-width: 1024px) > {}
/*LAPTOPS-DESKTOPS*/
@media (max-resolution: 131dpi) and (min-width: 1920px) > {}
```

Additionally, each one is usually duplicated for the portrait orientation as the landscape orientation is set by default. Moreover, css rules can be applied to screen with specific proportion (e.g. 16/9). Finally, since the release of high-definition mobile devices, the following tag must be added in the HTML head in order to communicate the browser how to adapt the content. This makes fit the width of the web page with those of the mobile phone.

```
HTML:
<meta name="viewport" content="width=device-width, initial-scale=1">
CSS:
/*SMARTPHONES*/
@media (min-resolution: 163 dpi) and (max-width: 966px) > {}
@media (min-resolution: 163 dpi) and (max-width: 966px and (orientation:portrait) >
{}
/*TABLETS*/
@media (max-resolution: 264dpi) and (min-width: 1024px) > {}
@media (max-resolution: 264dpi) and (min-width: 1024px) and (orientation:portrait)
> {}
/*LAPTOPS-DESKTOPS*/
@media (max-resolution: 131dpi) and (min-width: 1920px) > {}
@media (max-resolution: 264dpi) and (min-width: 1024px) and (aspect-ratio:16/9) >
{}
```

In order to optimize the code and the processing time, common features' values can be implemented only once before the Media Queries. Within the brackets developers may then modify specific values for specific displays.

There is infinity of possibilities that developers have to wisely manage without forgetting their goals: providing users the best user experience whatever device they are using.

Within their brackets, Media Queries allow developers to reorder, resize and reshape elements at some breakpoints while proportional dimensions resize automatically elements between two breakpoints providing adaptive contents, at least concerning the layout. Indeed, following sections will now focus on interaction design adaptations. These adaptations have to be specifically implemented in specific Media Queries in order to be adapted to the right device.

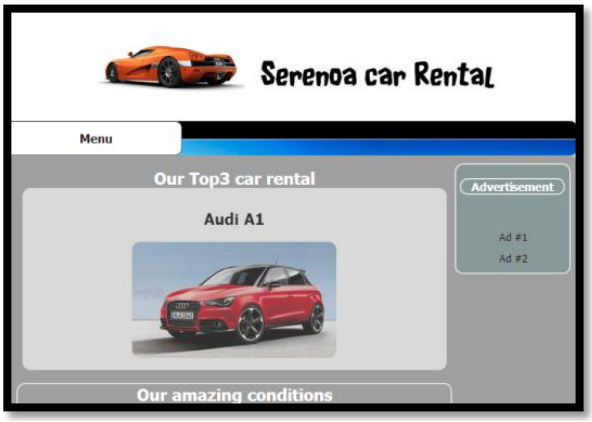
5.1.2.2. 'Mobile First' interaction designs

Mobile devices have reshaped the way users interact with the Internet. In the interaction design field of research, a device is mostly characterized by its screen, its inputs available and the way people use them. A recent survey stated 75% of users were using one thumb while browsing on their smartphones and two hands while using their tablets. Moreover, while a thumb can access every area of the screen, both hands easily access the sides of the screen (see figure 20 in appendix 1). Furthermore, smaller screens force developers to reshape primordial contents and layouts to fit the screen size.

Therefore, both layouts and interactions designs have to consider this reality. In following examples, these features are implemented on the car rental website on both tablet and smartphone versions. These interaction features will be detailed afterwards.

Menu accessible from the left side to embrace hands natural positions on tablets

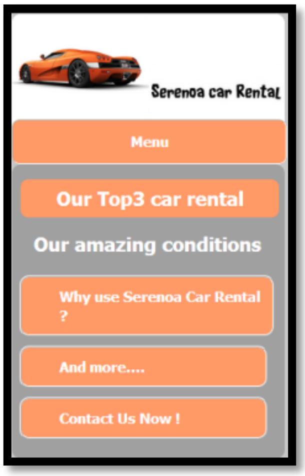
Initial tablet version
Orientation: landscape
Resolution: 620x586



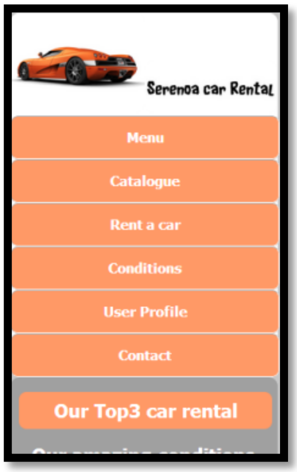
Tablet version: menu expanded
Orientation: landscape
Resolution: 620x586

Menu accessible from everywhere with thumbs on mobiles

Initial mobile version
Resolution: 320x480
Orientation: Portrait



Mobile version: menu expanded
Resolution: 320x480
Orientation: Portrait

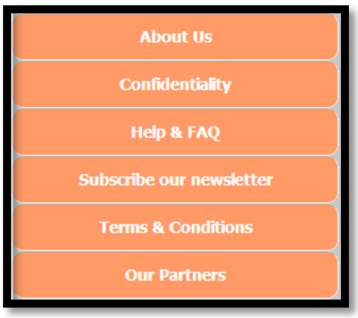


a) *Buttons vs links & the 'float' property*

Desktop-based computers have seen the proliferation of billion of clickable links that makes web browsing user-friendly and easier. However, these links have been designed for the small cursor of a mouse and not for thumbs. Embracing the mobile first approach, it is more efficient to design wider buttons instead of links as buttons are both easily clickable with a mouse and a thumb. However, buttons can be implemented from links with specific features. Indeed, by playing with CSS rules and Media Queries, developers can transform a button into a link into a fake button and vice-versa. Here is an example:

Different footers with adapted interaction designs

Mobile version
Orientation: portrait
Footer: 'fake' buttons



Desktop version
Orientation: landscape
Footer: links on desktops version

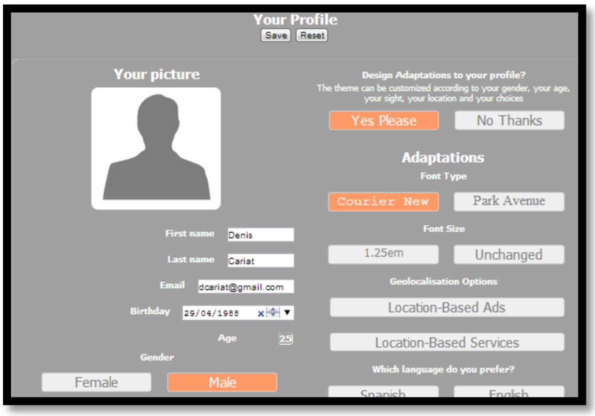

```
HTML:
<div class="footer" id="footer">
      <div class="lien" id="service">
            <a href=""><p class="services" id="service1"> About Us</p></a>
      </div>
</div>
```

```
CSS:
/*SMARTPHONES* border-radius uses absolute pixels and will render exactly
the same on every device/
@media (min-resolution: 163 dpi) and (max-width: 966px) > {
.footer .lien .services{width:100%;line-height:40px;margin:0%; float:none;
border-radius:10px;border-style: ridge; border-width:1px;border-
color:#FFF;}}

/*LAPTOPS-DESKTOPS*/
@media (max-resolution: 131dpi) and (min-width: 1920px) > {
.footer .lien .services{float:left;margin-left:3%; width:11%;}/*+colors
etc*/}
```

In the same way, radio buttons or 'ticks' may be transformed into wider 'fake' buttons. Here is an example with the 'user profile' page on the car rental website:

Desktop version
Orientation: landscape



Mobile version
Orientation: portrait

All these buttons are actually radiobuttons with specific CSS values.

```
HTML /*the text is inside the <input> and both are in the <label>. The all
becomes then clickable*/
<fieldset data-role="controlgroup"  data-role="fieldcontain" id="gender-
choice">
<legend><b>Gender</b></legend>
<label for="radio-choice-1"><input type="radio" id="radio-choice-1"
value="choice-1"/>
<span>Male</span></label></ fieldset>

CSS /*display:none; hides the 'radio' button*/
fieldset input{display:none;margin:auto;float:right;text-align:center;}
```

In previous pieces of code, one feature is a cornerstone in the implementation of cross-platform websites: the *float* property. While programming in HTML, elements are implemented under each other by default. In order to optimize the layout and use all the screen width on large screens, the *float* property allows elements to have floating elements on their sides.

Float property has to be adapted for each device

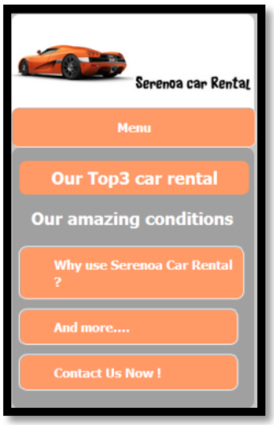
Desktop version Orientation: landscape	Mobile version Orientation: portrait
`img {width:50%;float:left;}`	`img {width:100%;float:none;}`

b) *Expandable contents*

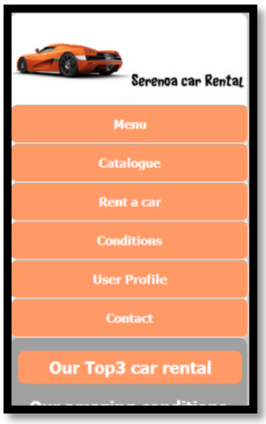
As previously explained, developers have to reshape contents in order to make them easily accessible in smaller displays. The idea is therefore to divide the content into several expandable contents accessible via clickable buttons or titles.

Contents are expandable according to users' needs

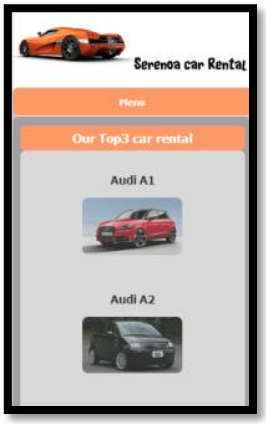
Initial mobile version



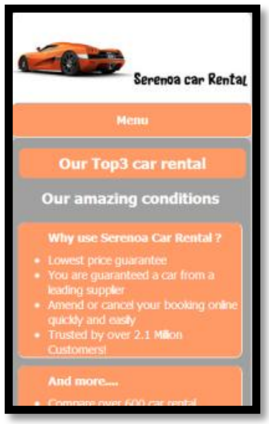
Menu expanded



album expanded



Text expanded



This may be easily implemented by setting the *onclick* function (e.g. *expand(div)*)to the specific expandable *div*. This function is a single *javascript* function modifying the *display* attribute of the specific *div:*

```
HTML:
<a onclick="expand(div5)"><p class="text_button"
id="text_button_6"><b>Menu</b></p></a>
<div id="div5" >/*CONTENT*/</div>
javascript:
function expand(object){ if
(object.style.display=='inline'){object.style.display='none'}
 else {object.style.display='inline';}}
```

c) *Drag & Drop contents*

Still in this purpose of using the available space to optimize the layout, developers could imagine contents that users may *'drag & drop'*. Conscious of this reality, HTML5 developers have included new 'Mouse events' or similar actions (e.g. click, double click) allowing actions when contents are dragged or dropped. Table 11 presents these new events.

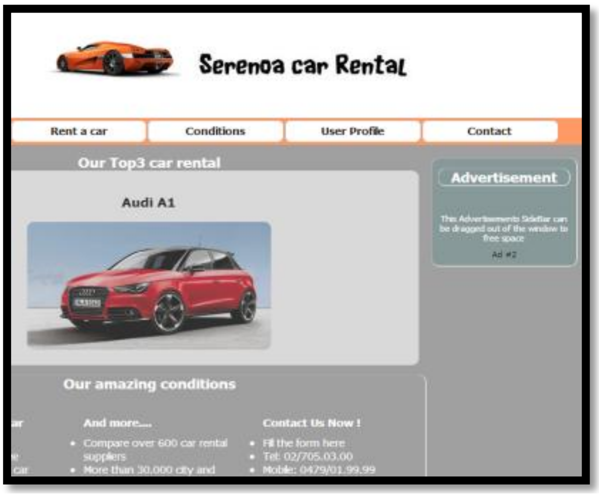
Events	Description
ondrag	"Script to be run when an element is dragged"
ondragend	"Script to be run at the end of a drag operation"
ondragenter	"Script to be run when an element has been dragged to a valid drop target"
ondragleave	"Script to be run when an element leaves a valid drop target"
ondragover	"Script to be run when an element is being dragged over a valid drop target"
ondragstart	"Script to be run at the start of a drag operation"
ondrop	"Script to be run when dragged element is being dropped"

Table 11 – HTML5's drag and drop events (W3SCHOOLS, 2013)

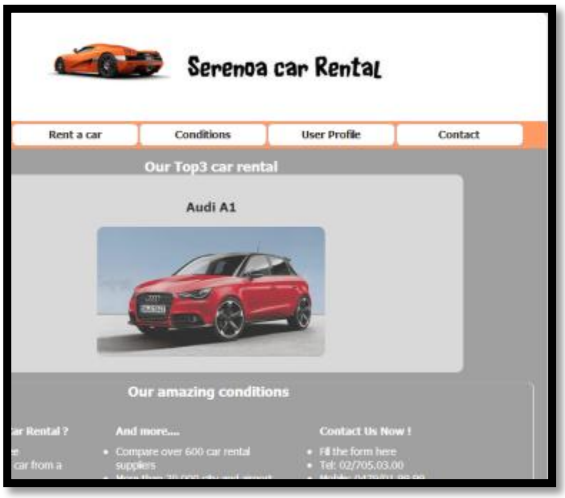
As the *onclick* event presented previously, linked functions have to be implemented in *JavaScript*. In that context, alongside with regular uses, users could invert elements positions or even drag them 'out of the window' to free available space for other contents. As an example, the car rental website has advertisements that may be hypothetically dragged out of the window:

Example of 'ondrag' event with the desktop version

Desktop version
Advertisements take 20% of the total width
(On the right)



Desktop version
Fluid contents use all the space available



Again, *JavaScript* function *hide_display ()* is straightforward and more elaborated function can be implemented:

```
HTML:
<div class="advertisements" id="vertical" ondrag="hide_display (this)">
JavaScript:
function hide_display (ad) {ad.style.display='none';}
```

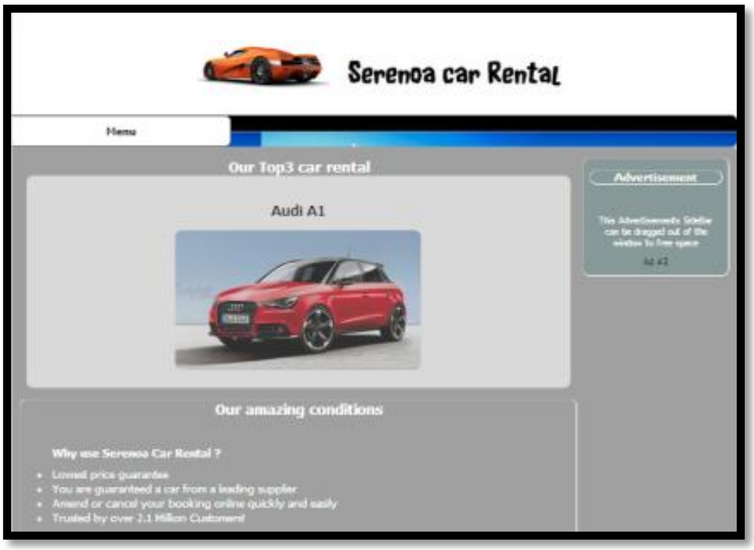
Finally, it is important to mention that these new HTML5's events are not supported yet on mobile browsers.

d) *Applications*

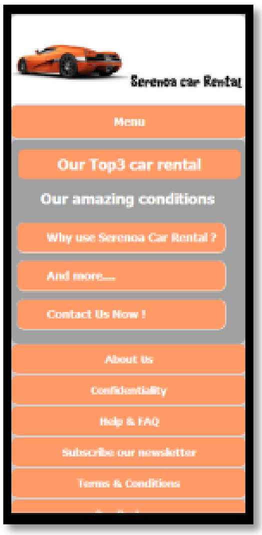
Previous concepts and techniques have been applied to the car rental website. This leads to following visual results:

Desktop version

Tablet version



Smartphone version



These key findings may be adapted to even larger screen. As depicted by the chart 27, wider is a screen, the more space is wasted.

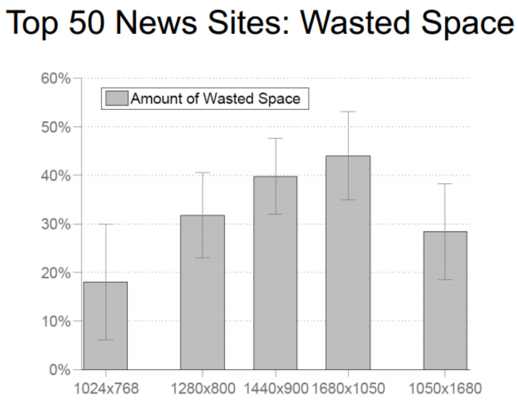


Chart 27 – wasted space in comparison with screen's size (NEBELING, NORRIE & MATULIC, 2013).

5.1.2.3. Downloading and processing time optimization

Now the content that users actually see (i.e. layouts and interaction designs) has been improved, developers can work on things that are invisible from users' eyes but crucial for their user experience: the downloading and processing speed. Nowadays, devices have different capabilities and while developers cannot improve themselves mobile devices capabilities, they are able to optimize the code or adapt the content to specific devices. With a combination of best practices and adaptations techniques, downloading and processing time may significantly drop. For this purpose, table 12 would help developers to numerically figure out what mobile devices are capable of while browsing through 3G or 4G networks.

	Megabits Per Second (Mbps)	KiloBytes Per Second (kBps)
3G	2	244
4G	6,2	756

Table 12 – 3G's and 4G's debit (SULLIVAN, 2012)

Key findings from the second chapter stated that 59% of users expect the loading time (i.e. downloading and processing time together) to be equal or lower than 3 seconds. With such expectations, files' size cannot be higher than 736KB for 3G or 2.268 KB for 4G. Unfortunately, downloading files is just the beginning. Indeed, once the files downloaded, the code has to be processed in order to display contents. Moreover, the processing time (e.g. around 80%) takes longer than the downloading time (e.g. around 20%). Finally, lower CPU and memory performances from mobile devices make users' expectations unreachable for the moment (JOHANSSON, 2013). For all these reasons, codes and contents have to be optimized to cut the global loading time.

a) *Adaptive images*

The *display* attribute is useful when it comes to expand or hide elements in real-time. However, this feature is not efficient or relevant when developers want to not display elements on some devices to avoid a long loading time (e.g. high-definition images on smartphone). Indeed, texts or images are still downloaded even if their *display* attribute is set to *none*. The conclusion is the same for the *visibility* attribute set to *hidden* (see figure 21).

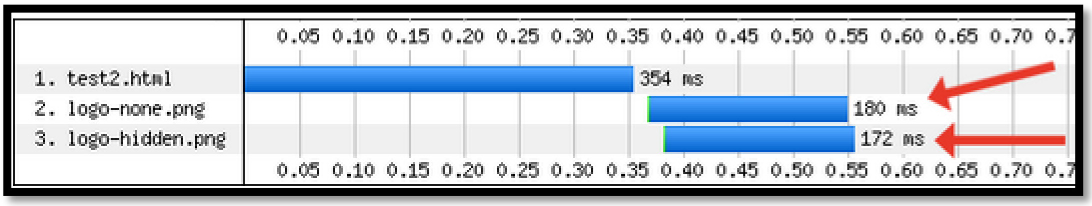


Figure 21 – Waterfall Chart for downloading time (JOHANSSON, 2013)

As images have significant sizes, they are the main component of the downloading and processing time and have to be studied particularly. Indeed, it would not make sense to waste resources to download an image 750 pixels wide for a smartphone screen of only 320 pixels wide. Therefore, images also have to be adaptive.

There exist two types of solutions for this issue. Either adapted images are downloaded within device-specific media queries or “there are solutions based on JavaScript where specific images can be fetched from the server depending on screen size” (NEBELING & NORRIE, 2013). Pro’s and con’s for each will be highlighted before giving trends for the near future.

On the one hand, using CSS and Media Queries to download adapted images on specific devices works (GRISBY, 2010) but may appear quickly daunting. Indeed, for each picture to display, developers have to resize it as much as there are media queries. Then, each version has to be correctly coded in the corresponding Media Query.

```
HTML:
<div class="image"></div>
CSS:
@media (max-width:600px){image { background-image: url(image1-low.jpg);}}
@media (min-width:601px) {image {background-image: url(image1-high.jpg);}}
```

As a result, CSS code will be bigger with the already-known consequences on the downloading and processing time.

More convenient solutions are under discussion within the W3C Responsive Image Community Group to create a new element <picture> using the same syntax as the <video> element. Similarly, each image versions would be referenced separately in a sub-element <source> corresponding to a specific Media Query (RESPONSIVEIMAGES.ORG, 2013).

```
<picture width="500" height="500">
 <source media="(min-width: 600px)" srcset="large.jpg">
 <source media="(min-width: 200px)" srcset="medium.jpg">
 <source srcset="small.jpg">
 <img src="small.jpg" alt="">
 <p>Text Available</p>
</picture>
```

Alternative solution would be keeping the <img> element but giving it a new attribute called *srcset*, used as following (RESPONSIVEIMAGES.ORG, 2013):

```
<img alt=""  src="banner.jpeg"
 srcset="banner-HD.jpeg 2x, banner-phone.jpeg 640w, banner-phone-HD.jpeg
640w 2x">
```

Each url is separated by a coma and additional conditions can be added such as the viewport maximum size or the device pixel density.

On the other hand, leaving the adaption task to the server may look more convenient. Nowadays, some developers propose such solutions based on *JavaScript* as *Adaptive Image* “which detects your visitor's screen size and automatically creates, caches, and delivers device appropriate re-scaled versions of your web page's embedded HTML images” (WILCOX, 2013). Even thought it looks convenient, it uses JavaScript and some requirements limit the scope of such solution. Indeed, *Adaptive Image* has to be run on Apache Server and requires PHP5. Unfortunately, even if popular, these technologies represent respectively 65% and 80.6% of market share (W3TECHS.COM, 2013). In addition to the fact that users can turn off JavaScript, such solutions fail therefore at becoming a standard. Indeed, “an overreliance on scripts and server applications would lead to additional complexity and redundant HTTP requests to the development process. Furthermore, script-based solutions will be unavailable to users who have turned off JavaScript” (RESPONSIVEIMAGES.ORG, 2013).

As a conclusion for this section, it looks like the most convenient solution (e.g. <picture> element or *srcset* attribute) will come from the W3C. As mobile devices and their lower capabilities spread over the world, it would make more sense to implement a standardized browser-based solution in HTML5 as it already exist for <video> elements.

b) *Programming best practices*

Once developers have implemented a multi-platform website as well as managed the images issue, the development may be over at this point if they followed a set of programming best practices during the development. This section would have come at the beginning but it is more relevant to develop it now as previous contents and the rental car’s website can be used to illustrate it. These code optimizations decrease files' size and therefore diminish both the loading and the processing time. In its developers' tools, Google offers a free extension called Page Speed which sums up all the best practices that developers may also find everywhere on the Internet. This tool "evaluates pages conformance to a number of different rules" (GOOGLE, 2013) which were already available and meaningful before the first mobile device. These guidelines focus on six categories of practices: "optimizing caching, minimizing round-trip times, minimizing request overhead, minimizing payload size, optimizing browser rendering and optimizing for Mobile" (GOOGLE, 2013). Alongside more technical recommendations, some concern images, CSS and JavaScript files allowing a lower loading time:

- **Combining external scripts and style sheets** "into as few files as possible cuts down on RTTs (Real Time Technologies) and delays in downloading other resources";
- "Combining images into as few files as possible using **CSS sprites** reduces the number of round-trips and delays in downloading other resources, reduces request overhead, and can reduce the total number of bytes downloaded by a web page";
- "**Correctly ordering external style sheets and external and inline scripts** enables better parallelization of downloads and speeds up browser rendering time. Because JavaScript code can alter the content and layout of a web page, the browser delays rendering any content that follows a script tag until that script has been downloaded, parsed and executed";
- "**Setting an expiry date or a maximum age in the HTTP headers for static resources** instructs the browser to load previously downloaded resources from local disk rather than over the network";
- "**Compacting JavaScript, HTML and CSS codes** can save many bytes of data and speed up downloading, parsing, and execution time";
- "**Deferring loading of JavaScript functions** that are not called at startup reduces the initial download size, allowing other resources to be downloaded in parallel, and speeding up execution and rendering time";
- "Properly **formatting, compressing and scaling images** can save many bytes of data" as well as specifying the right image dimensions;

As final point for this section, here are the recommendations made by Google Page Speed extension for the car rental website's main page, not optimized on purpose. For each recommendation, computed gains in size (%) and KB are provided. This allows to figure out to which extend optimizations are important for bandwidths and the processing time.

Recommendation	Gains in %	Gains in KB
Enable compression	76%	78.1
Serve scaled images	53%	71.4
Optimize images	9%	12.5
Minify CSS	49%	12.3
Minify JavaScript	32%	8.7
Minify HTML	19%	1.0
Specify image dimensions	No figure	No figure
Defer parsing of JavaScript	No figure	No figure
	TOTAL	184 KB
	TOTAL TRANSFERED	416.02 KB
	GAIN %	44%

Table 13 – Gain from programming best practices

This represents a gain of 0.75 second for 3G connections and 0.24 second for 4G connections. These figures are significant as they do not take into account the processing time. Finally, it is relevant to also point out that the car rental website is straightforward with only four low-resolution images on the main page.

5.1.3. Key findings

Techniques	**Pros**	**Cons**
Proportional layouts	Fit to the screen's width between two media queries	May not perfectly fit (e.g. too stretched) if screen's width is unconventional and not enough media queries implemented
Media Queries	CSS rules may be changed for different screen's width or other metrics. Avoid the creation of several websites	Centimeters and inches are not reliable as pixel densities vary across devices
Buttons vs links	Easily clickable on every device	May disturb users if links are not used anymore
Expandable contents	More convenient on smaller screens	Not working if javascripts disabled
Drag & Drop	More convenient and more intuitive on touch screens and increased accuracy for people with tremor. Layout customizable.	Not yet implemented in mobile browsers. Not working if javascript disabled
Adaptive Images (javascript)	Gain in performance	Not working if javascript disabled. Longer processing time on both server's and users' sides
Adaptive Images (new HTML5 element)	**Working** if javascript disabled More gain in performance (no javascripts) Easy to implement	More work on developers' side (different version for each picture)
Programming best practices	Gain in downloading and processing time	Time and resources consuming if not implemented from the beginning

Table 22 – Key Findings for the platform dimension and related adaptation techniques

5.2. Dimension #2: the surrounding environment

Since the emergence of mobile devices, information systems and websites have left homes, offices and laboratories to spread all over the world. Furthermore, thanks to wireless technologies (e.g. wifi, 3G, 4G connections), these devices have become even more used than desktops or laptops to access websites or applications for the simple reason that these technologies embrace the way users live. Thanks to mobile devices' location awareness, location adaptations are therefore relevant to help and advise people in their everyday life. Now the first layer of adaptations (i.e. cross-platform adaptations focusing on layout and interactions) has been implemented, developers may adapt the content and add adaptations related to the current position (i.e. absolute position) or even related to the surrounding environment (i.e. relative position). Time dimension is related to location dimension. Indeed, if servers know where users are, they also know what time it is. The functioning of this technology, applications and usage statistics are described in this section.

5.2.1. How does it work?

HTML5's developers have implemented the Geolocation API which allows users to share their location. In exchange, they subsequently receive location-based services, features and advertisements (e.g. position on a map, close businesses) as longitude and latitude may be afterwards used in javascript thanks to this API. As depicted by table 14, other information can be extracted as the altitude, the accuracy, the speed and the orientation towards the true north.

Property	Notes
coords.latitude	decimal degrees
coords.longitude	decimal degrees
coords.altitude	meters above the reference ellipsoid
coords.accuracy	meters
coords.altitudeAccuracy	meters
coords.heading	degrees clockwise from true north
coords.speed	meters/second
timestamp	like a Date() object

Table 14 –information provided by the Geolocation API (W3SCHOOLS.COM, 2013)

More accurately, GPS, wifi and 3G-4G's networks' functioning are a bit different. While location of devices using wifi or cables can be located with their IP address, GPS and devices using 3G or 4G connections are detected thanks to a method called *triangulation* (see figure 22). At first, devices send signals to the closest base stations (or satellites for GPS). Subsequently, triangulation calculation based on this information is performed by positioning algorithms and software. Finally, using other algorithms and tables, accurate geographical location can be provided with an error rate expressend in meters.

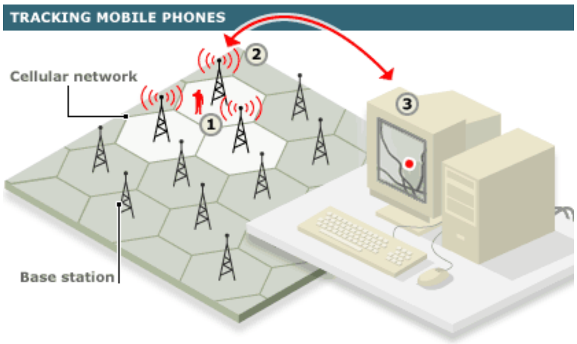


Figure 22 – Triangulation (BBC, 2005)

However, it would not be possible without the assumption that users have authorized websites to access their location. The feature is supported by all browsers and can be set in the settings panel. Even if users turn off geolocation, they are able to set some exceptions, for their preferred websites for instance. Javascripts also need to be activated as “The Geolocation object is used by scripts to programmatically determine the location information associated with the hosting device. The location information is acquired by applying a user-agent specific algorithm, creating a Position object, and populating that object with appropriate data accordingly” (W3.org, 2013). Before giving examples of Geolocation adaptations, the next section highlights location-based adaptations that can be operated thanks to other sensors embedded in recent devices.

5.2.2. Surrounding environments’ conditions

Even if geolocation may be used to determine the local weather, meteorology is not an exact science yet and local brightness or other local features (e.g. acoustics) cannot be detected through the API. In that context, it is then rational that recent years have seen the release of dozens of applications regulating automatically the brightness of the screen (e.g. on Google Play application market). Moreover, recent devices – mobile or stationary – include an auto-brightness control feature by default trough embedded sensors. Ideally, display’s brightness has to be similar to the surrounding environment display. "If it looks like a light source, it is too bright. If it seems dull and gray, it may be too dark” (HEITING, 2013). Indeed, this would be redundant and ineffective that every websites adjust the brightness of the screen, brightness varying from websites according to the algorithm implemented. In that context, screen brightness is automatically regulated according to the light measured by the sensor resulting in an increased productivity (e.g. eye strain diminution) and energy savings.

Concerning environment acoustics, some applications already exist regarding speech recognitions or safety issues such as sound detectors. Further, applications like *Shazam* allow user to know songs' title through the built-in microphone and specific algorithms. However, these applications are used for specific tasks and only react in presence of specific events. Indeed: "Research on general unstructured audio-based scene recognition has received little attention as compared to applications such as music or speech recognition" (CHU et al., 2006). For this purpose, fundamental researches "content analysis for acoustic environment classification in mobile robots" had been conducted in 2006 by CHU et al. Their purpose was to recognize the surrounding environment and learning from it. Further, they investigated into possible adaption features and the feasibility of such system. These researches are described as they obtained relatively satisfying and accurate results. Here below are descriptions and conclusions.

Their first motivation was that "A stream of audio data contains a significant amount of information, enabling the system to capture a semantically richer environment" (CHU, 2006). Moreover, in opposition to vision-based systems, audio-based systems do not need specific conditions to work efficiently such as a sufficient lightning and are also cheaper than visual recognition software and algorithms. However, both visual and auditory technologies have to be considered together to obtain accurate results but this research bases the recognition on acoustic information only in order to be able to detect environment commonly encountered by users. Here below are the steps researchers went through showing how such environment characterization and related adaptations can be implemented.

The first step is characterized by the gathering of sound samples representing auditory common environments. They firstly restricted the study to 5 different environments: streets, elevators, cafés, hallways and lobbies. Subsequently, audio features (e.g. energy and spectral moments) are analyzed to characterize each environment (see figures 23 and 24).

Class	Energy (x10-4)	Energy Range	Zero-Crossing
Street	0.145	0.919	14.19
Elevator	0.123	0.917	18.60
Café	0.048	0.866	29.99
Hallway	0.062	0.977	11.90
Lobby	0.064	0.932	29.09

Figure 23 – Noises' energy per class (CHU & al., 2006)

Class	Frequency Range (Hz)	Mean (Hz)	Mode (Hz)	% of data w/ mode value
Street	0-172	74.5	86	26.7
Elevator	0-172	72.1	43	43.3
Café	0-603	178.6	129	66.7
Hallway	0-172	66.0	58	64.4
Lobby	0-560	164.3	129	56.7

Figure 24 – Noises' main features per class (CHU & al., 2006)

Thirdly, researchers described evaluation data corresponding to everyday auditory environments which allows afterwards considering context-aware applications through sounds recognition. Finally, supervision learning is applied in order to classify new samples in the right category. As a result, they showed that "even from unstructured environment, it is possible to predict with fairly accurate results the environment that the robot is positioned" (CHU et al, 2006). They stated that even if only five different environments were analyzed, accurate recognitions were possible only on their general auditory features. With such results, they easily imagine applications able to regulate volume automatically. Further, in combination with geolocation, applications would be able to determine where users are and what they are actually doing (e.g. if a user is close to a specific restaurant, sound recognition would allow to confirm that this specific user is actually in the restaurant). From that, restaurant's owner may allow discount to people actually in the restaurant. Finally, mobile devices' volume can be automatically turned off while users are eating and turned on when they live the restaurant. As stated by researchers: "This current work opens up a doorway to other open challenges".

5.2.3. Language Detection

As developers are able to know users' location thanks to different means, they are therefore also able to infer the local language from the current location. Here below are the most common techniques to determine the language. They all have pro's and con's which and each better fits with specific types of websites and applications (GUEMBEL, 2012). And even if language's adaptations are useful mainly for global websites of multinationals, it can also be useful for companies or institutions lying in a multilingual country as well as for NGO's.

5.2.3.1. Detection based on the IP

Every device able to access the Internet owns a different IP address (Internet Protocol) which is either permanent or temporary. Thus, IP address as to be extracted from the HTML request and also has to be matched with the location with corresponding latitude and longitude. There are some underlying implications while using this technique. At first, developers often have to implement some pieces of code on the server that have to be linked with databases either free and often outdated or commercial and accurate. Last but not least, the provided location may be wrong as Internet access may first go through VPNs, Proxy's or non-local Firewall.

5.2.3.2. Detection based on HTTP Header

Another option is looking at what is written in the HTML header's request with PHP request such as:

```
PHP:
$_SERVER["HTTP_ACCEPT_LANGUAGE"])
```

It actually extracts users' language preferences from the browser's settings panel. Unfortunately, even if more accurate, it does not indicate anything on current users' location. Finally, English may be set by default by users in many small countries whereas their mother tongue is different and rarer.

5.2.3.3. Detection based on HTML5 Geolocation API

HTML5's new features integrates the Geolocation API to respond to growing demands and supplies of location-based services and features induced by mobile devices' success. It works remarkably under one crucial assumption: users have to confirm or accept by default requests for geolocation and javascripts. One the one hand, no issues rise when it comes to provide the way to closest hotel but it may seem intrusive many other cases. Discussions about privacy issues will be held in the last chapter.

5.2.3.4. Keep it simple: ask the user

In front of such issue, Occam would say: "ask simply users which language they prefer". And that is what some websites do. Indeed, on travel websites or air companies' websites, it would not make sense to assume users' current location reflect their homelands or their favorite languages. As a result, it is therefore more relevant to simply ask users which language they prefer. For instance, if users access the Internet from the reception's computer, language detection based on the previous techniques will all lead to wrong assumptions.

5.2.3.5. A website for each region

In order to solve language issues, a trivial solution consists in duplicating websites in several languages and corresponding websites' addresses. In that context, developers can provide adapted websites in different regions. However, this technique has several undesirable consequences. At first, unless the audience is strictly limited to a small the number of languages, it induces to develop and maintain a plethora of different websites. Subsequently, multiple web addresses with the ‘same’ target is bad practice for Search Engine Optimization.

5.2.3.6. Best practices

As there is a multitude of different languages and as users’ mobility increases, a vast number of different and unlikely situations may occur. For that reason, best practices recommend to ask the user to confirm the language that has been detected, whatever the technique used. If not the preferred one, users must have the option to modify it. Another way to confirm the preferred language is using two or more language detection techniques.

5.2.4. Location-Based features, advertisements and social Networking

These features are unarguably the most representative of the worldwide expansion of mobile devices. “With over 770 million GPS-enabled smartphones, location data has begun to permeate the entire mobile space. It's powering advertisements, and many other services — from weather to travel app” (BUSINESS INSIDER, 2013). Location detection techniques have been described in the previous section. However, GPS systems are now embedded in recent mobile devices and facilitates the location process. Roots of these features come from the emergence of internet technologies which set users and their devices in motion. Since 2010 which “is proven to be a big breakout year” (VAN GROVE, 2010), developers and companies have directly seen the huge underlying impact it may have on implemented features as well as related revenues and brandawareness they could get from location-based features. Without any surprise, many surveys and researches state that location-based features’ public adoption has followed the increasing adoption of mobile devices. But previously, a quick overview of popular location-based features is necessary. A large list of location-based services is available in the appendix 6.

Recent years have seen the apparition of a plethora of location-based services allowing users to recommend places or events, to search for nearest service providers (e.g. ATM), to locate someone or ‘check-in’ at some places (e.g. Facebook, Instagram, Twitter, Foursquare), to receive location-based advertisements, to recover lost items, to play games in which users’ location is part of the game for instance. As in any area, some best practices are recommended while implementing such feature (WEB BUSINESS, 2012), especially for location-based marketing which may use a several technologies to reach potential clients such as Bluetooth Marketing (i.e. users receive messages when they pass by a store), near field marketing (wireless communication on very short distance, between a card and a card reader for example) and location based services and advertisements. At first, consumers will tend to avoid location-based advertisements as they are concerned by privacy issues as well as the fact that these technologies are still in their infancy and not completely integrated yet into users' everyday life. So, such features and opt-in instructions have to be clear in order to avoid confusion. Secondly, flooding users with notifications, recommendations, messages or offers will lead to many users' opting-out. Thirdly, location-based features have to be tested on unbiased groups of users. This means testing these features with users of different age, gender, ethnicity, language and according to their revenues for instance. Finally, practice has shown that rewarding users is a good way to make users opt-in and come back.

Now the scene has been set, why is it so important for businesses and organizations to embrace this trend? The second chapter highlighted the ascendance of mobile devices into users' daily life with an important but not complete substitution effect from desktop computers. Therefore, not implementing such features means many undesirable effects for businesses: smaller market shares, narrower brand awareness and subsequently less revenues and profits. Chart 28 highlights the most relevant trends, showing how these location-based features become predominant in users' everyday life and there crucial for businesses and organizations.

Chart 18 highlights the attractiveness of such features and the nearness' link between businesses and customers. It states that the closer users are to a business, the more likely they are to click on a mobile ad for that specific business.

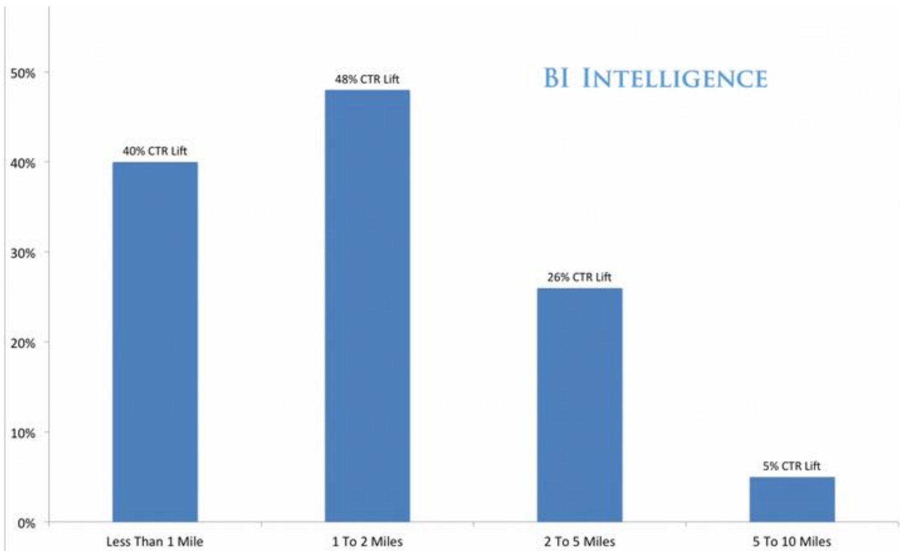


Chart 28 – Link between proximity and ads (BI INTELLIGENCE, 2012)

Moreover, other statistical researches state that already in 2012, "74% of smartphone owners use their phone to get real-time location-based information, and 18% use a geosocial service to 'check-in' to certain locations or share their location with friends" (ZICKUHR, 2012). As these technologies are young, the most remarkable fact is the quick rise of these technologies in a short period of time, following the increasing smartphone ownership's rate. For instance, geosocial services and location-based information services have respectively reached 18% and 74% in use for users who own a smartphone (see chart 29).

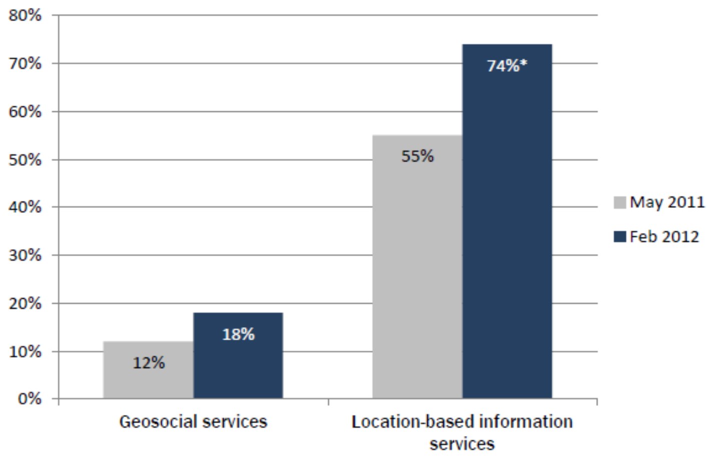


Chart 29 – Geosocial services' and location-based information services' adoption (Zickuhr, 2012).

Further, these data also show discrepancies in use between users from different age, gender, ethnicity, household income and education level (see table 15 in appendix 2). These characteristics must therefore be taken into account while implementing location-based features.

While genders and ethnicities do not seem to be relevant, the age, the household income and the educational level seem to be relevant for specific market segmentation analysis. Further results are provided in location-based features are already widely spread and the focus will therefore be set on the next dimension: the user.

5.2.5. Key findings

Feature	Techniques	Pros	Cons
Brightness regulation	Embedded sensor	Eye strain diminution. Battery savings	May lead to undesirable modifications
Environment recognition	Embedded microphone	Many adaptations possible: volume control, increased location accuracy, adapted services	Categorization may lead to errors and oversimplification in specific environments
Language detection (mix them for a better performance)	IP	Easy to implement	May rely on outdated (and not free) databases Users may use proxy's or VPNs
	HTTP Header	Extraction from browsers' settings	Language may set on English by default in smaller countries
	Geolocation API (HTML5)	Straightforward	Need geolocation and javascripts enabled. Privacy issues
	Ask the user	Avoid errors. Fit perfectly with travel websites for instance	Not context-aware
	Regional websites	Region-adapted websites	Maintenance of several websites. Bad for Search Engine Optimization
Location-based features		Many adaptations that respond to users' needs and businesses aspirations	Privacy issues. Need geolocation and javascripts enabled.

Table 23 – Key Findings for the environment dimension and related adaptation techniques

5.3. Dimension #3: the user

For decades now, researchers in computing and related fields have dreamed about devices and applications that adapt themselves to the user instead of the opposite. As a starting point to reach this feat, websites and applications have to already integrate features that meet users' common expectations regarding adapted layouts, user-friendly interaction designs and location-based features. Once correctly implemented, developers may start thinking about user-specific adaptations. Indeed, as a third layer of adaptations, this has therefore to be implemented on a solid basis.

A convenient way to structure this section is to dissect users' characteristics trough their interaction with current devices. Assumption is made that information systems are already adapted regarding the first and the second dimensions. In order to interact with systems, users must 'see' at first how the application is displayed. Through the sight, discrepancies appear between users (e.g. font types' and font sizes' preferences, myopia, presbyopia, color-blindness, blindness) and lead to misunderstandings and misconceptions. Subsequently, in the interpretation phase, other users' characteristics appear such as their age, their gender, their personality, their current mood and the current activity. Finally, during the actual interaction with the system, sight's issues appear again and additional users' characteristics may emerge and disturb the interaction (e.g. tremor).

Through the consideration of these issues, other challenges rise such as users' uniqueness as well as the way to collect and interpret these changing features. Fortunately, today's browsers allow to collect some users' default options (e.g. font type, font size). Further, interpretations of users' uniqueness can be made through a categorization process but the real challenge lies in the way to capture specific personal information (e.g. users' mood?). Moreover, assuming such information are easily collectable, some are constant (e.g. gender) or measurable (e.g. age) while others change constantly (e.g. personality, current activity, mood) and this would produce an enormous amount of data, surpassing current devices' capabilities.

On the one hand, this section highlights adaptations techniques that may be implemented in order to improve user experience. Examples of such adaptations and existing tools are presented. On the other hand, as debates and researches are currently in progress, this chapter provide insights in user categorizations and in different ways of collecting changing personal information.

5.3.1. Adaptations techniques

5.3.1.1. Age and Common sight troubles

From their birth or as they are aging, majority of users present sight troubles in many different forms. As the open window on the world and therefore on how information systems are displayed, eyes and their common troubles have to be analyzed and assessed. While some troubles are common and arise as users are aging, they still affect the interaction between users and systems. Others are less common and have bigger impact on users' interaction with systems. The majority of these troubles may be managed with simple means (e.g. glasses). However, it is worth to be studied as the Internet is now accessible anywhere at any time and people do not wear these corrective means all the time (e.g. oversight, leisure time).

Table 16 lists these troubles, related effects and the common age of apparition:

Age	Sight Troubles and Diseases	Consequences on user experience
40s	People cannot escape presbyopia	Difficulty with near vision focus
	Increasing risk of dry eyes	Blurred vision, contact lenses are uncomfortable, "decreased tolerance of reading and working on the computer" (National Eye Institute, 2013).
	Increasing risk of computer vision syndrome	Eye strain, decreased productivity
50s	Presbyopia becomes more advanced as well as the risk of dry eyes for women after menopause	
	Increasing risks for cataracts, glaucoma & macular degeneration	Blurred vision, colors may not appear as they really are, light may be glaring
60s	Increasing risk for common eye diseases (see previously).	
	Lower ability to see in low lighting	Eye strain, decreased productivity
70s+	Cataracts troubles are common and color vision declines.	Previous consequences increase in intensity

Table 16 – Sight troubles: age and consequences (HEITING, 2010)

Taking these troubles into account while designing a website would necessarily improve the user experience for some users, mostly those aged of 40 or more. Indeed, these adaptations would be relevant as older people become more and more tech-savvy. However, younger users can also be affected. While recent screens are anti-reflective and are more comfortable for the eyes, others features must be managed by developers in order to avoid users to adjust them by themselves (i.e. brightness, text size, contrast and color temperature). Here are insights in these features.

Ideally, display's brightness has to be similar to the surrounding environment display. "If it looks like a light source, it is too bright. If it seems dull and gray, it may be too dark"

(HEITING, 2013). However, this would be redundant and ineffective that every websites adjust the brightness of the screen, brightness varying from websites according to the algorithm implemented. In that context, it is then rational that recent years have seen the release of dozens of applications regulating automatically the brightness of the screen (e.g. on Google Play application market). Moreover, recent devices – mobile or stationary – include an auto-brightness control feature by default trough embedded sensors. Therefore, the brightness issue seems to be managed even if improvements are needed as many Internet surfers ask how to disable this feature. Besides the brightness, the focus will therefore be set here on issues that devices' manufacturers cannot tackle. Those that developers should consider while developing websites or applications: text size, contrast and color temperature according users' characteristics. WHEILDON (2005) led deep researches in that field and conclusions of his book *Type & Layout* will be provided.

a) *Font Type and Font Size*

As sight troubles seem to be inevitable from the age 40, it makes sense for developers to adapt font types and font sizes in order to be readable by everyone visiting their websites.

Concerning font types, it may be more important than it looks like: "It's possible to blow away three-quarters of our readers simply by choosing the wrong type. If you rely on words to sell, that should concern you deeply" (WHEILDON, 2005). Key findings are surprising: best font types for printed books are not the same for online supports.

For printed versions, key findings are unarguably: serif font types (e.g. Times New Roman, Garamond, Georgia, Courier) are more than five times more readable than sans serif font types (e.g. Helvetica, Calibri or Arial) and "those who read the sans-serif version said they had a tough time reading the text and continually had to backtrack to regain comprehension" (WOOD, 2011). For online supports, sans serif fonts have seemed to be more legible. As font details are proportional to the number of pixels a screen is able to display, sans serif font render better than serif fonts which are more detailed and require then a higher resolution display. Here are conclusions from a research conducted in 2002 when the best screen resolution was 800x600 pixels: "for easiest online reading, use Arial 12-point size and larger. If you're going smaller than 12 points, Verdana at 10 points is your best choice. If you are after a formal look, use the font Georgia. And for older readers, use at least a 14-point font" (WOOD, 2011). However, screen resolutions have increased since then, allowing a better rendering of serif font types on screens and then blurring boundaries between printed and online versions.

As a conclusion, it is therefore highly recommended using sans-serif font types for online supports which are more readable on screens (see figure 25 in appendix 1). However, when it comes to print this support, serif font types are recommended for body text and sans serif font types may be used for headings or in order to emphasize some sections. Finally, browsers allow users to set their favorite font type which are applied if no font types are specified in the CSS code of a specific website. However, for obvious reasons of rendering, font types are usually determined in the CSS file (i.e. a specific sans-serif font size with the fallback option ‘sans-serif’ if chosen font types are not supported by the browser). The developer is therefore responsible for the website’s legibility and no issue will emerge if font types are carefully chosen.

```
CSS:
body{font-family: Arial, Verdana, sans-serif;)
```

Concerning font sizes, even if a size around 13 points is commonly used for body texts and 24-26 points for headings within web developers (MARTIN, 2009), it has been proven than a larger font type is preferable for users with sight troubles, at least between 16 and 18 points for body texts but headings can remain unchanged if large enough. However, from an aesthetical point of view, it may be inconvenient to make the font size larger for everyone. As a result, font size should be adapted in function of users’ age and their sight troubles. Nowadays, browsers allow users to configure their favorite font size through the setting panel, instead of zooming in on every web page. However, it works only on website in which proportional font sizes are implemented. Unfortunately, many websites still use fixed font sizes. Here are some pieces of code to illustrate the difference with the assumption that users have already set up a default font size of 18 points in their browser:

CASE#1: all font sizes are fixed. The website will be rendered with these fixed sizes without taking into account browsers’ default font size

```
CSS:
body {font-size: 13px;}
h1 {font-size: 18px;}
```

CASE#2: body font size is proportional to browser’s default and others are fixed. Only body text’s font size will be adapted with the assumption than headings are eligible enough.

```
CSS:
body {font-size: 100%;}
h1 {font-size: 18px;}
h2 {font-size: 16px;}
```

CASE#3: body font sizes using ‘em’ are proportional to browser’s default and others are proportional to the parent (i.e. <body>). Body text’s font size will be adapted and others will be proportional to body’s font size.

```
body {font-size: 100%;}
h1 {font-size: 1.4em}
h2 {font-size: 1.2em}
```

If default size is 10px, it means 10px for the body, 14px for h1 and 12px for h2.

The last solution (i.e. case#3) seems to be the most convenient as everything is proportional to browser’s default font size. The second solution is also convenient under the assumption that fixed font sizes are large enough (i.e. at least between 16 and 18 points). The first solution is not adaptive and must be forgotten. With the assumption that font sizes’ preferences are set up in advance by the user in its browser, developers are again responsible for the website’s legibility.

b) *Font Color, Contrast & Color temperature*

The majority of developers seem to have opted for almost pure white background with dark body and headings (Martin, 2009). This seems to have reached a consensus which is the most convenient solution for older readers or readers with partial sight (ARDITI, 2013). Indeed, dark letters on clear backgrounds or the opposite are known to be the most readable combination. However, comprehension seems to be altered by the second combination: “When text was printed black on white, readers reported good comprehension 70% of the time, fair comprehension 19%, and poor comprehension 11% of the time. When text was printed white on black, good comprehension fell to ZERO, while poor comprehension rose to 88%” (WHEILDON, 2005). It therefore can be used for irrelevant information. Further, for aesthetical reasons, combinations of colors can be operated but the rendering will never be as readable as a black-white combination. “Even a seemingly innocuous background color such as pale blue has pretty dire consequences” (WHEILDON, 2005).In that context, it is highly recommended to use a combination of colors with a sufficient contrast to be readable unless developers are able to know users’ age (e.g. websites requiring users’ identification). If aesthetics influences more the layout than the user experience (e.g. a specific brand) developers may modify background and font color according to users’ age

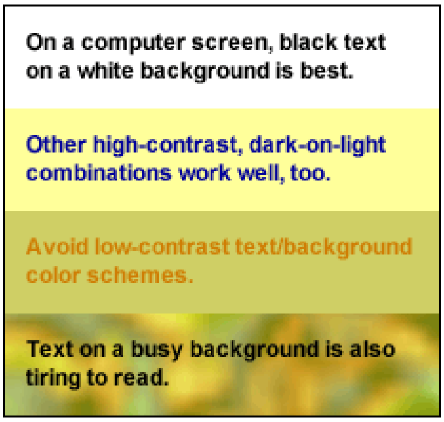


Figure 26 – Contrast matters (HEITING, 2013)

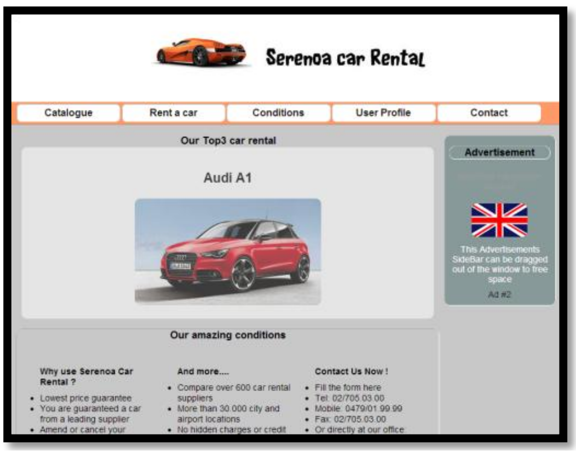
through JavaScript function. A straightforward example is provided in the next section.

5.3.1.2. Applications

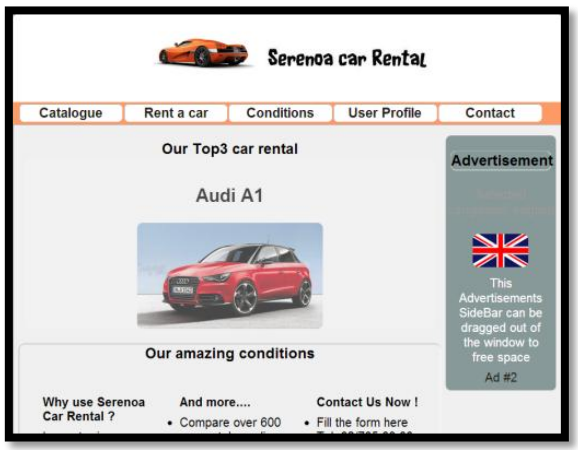
With previous recommendations, users' experience may reach unprecedented levels if both parties do their work (i.e. users and developers):

On the users' side, preferred font type and font size have to be set once in the browser's default settings. Moreover, for those suffering from color-blindness, it is highly recommended to set up specific extensions. On the developers' side, font types have to be carefully chosen and font sizes have to be proportional to users' preferences. Moreover, developers have to pay attention to color contrasts and color temperatures in order to take into account elderly users and their decreasing visual acuity.

Following assumptions are made for the following example: the first user is 30 years old and has set its favorite font type to 'Times New Roman' and its font size to 16 pixels. The second user is 70 years old and has set its favorite font type to 'Verdana' and its font size to 22 pixels. Both are not color-blind. Finally, the developer has implemented font type to Arial and proportional font size to 100% for the body text, 1.2em for headings (h1) and 1.1em for headers' and footer's buttons. Finally, black letters have been chose in combination with a light grey background. Assumptions is also made that users' age is available through the database. Here are automatic results:



Redering for user#1 (30years):
Grey background a regular font size (by default)



Redering for user#2 (70years)
Higher contrast and larger font size

Users' font types are not taken into account as the developer set the font type to Arial. Except this, all other users' features have been taken into account without any additional effort from both parties. Here is how the age is simply taken into account through JavaScript in order to modify content's background and text's size:

```
JAVASCRIPT:
      if (localStorage.age > 40)
{var y=(170+localStorage.age);
      document.getElementById('content').style.backgroundColor ='rgb (' + y
+ ',' + y + ',' + y +')';}
```

‘localStorage’ is a new HTML5 feature supported by all browsers and intended to replace the use of cookies. The value of ‘localStorage.age’ may be set when user logs in. Unfortunately, assumption is made that users under 40 have perfect visual acuity and users above 40 always have sight’s troubles. Therefore, adaptations should require validation from users to avoid useless adaptations.

5.3.1.3. Other troubles

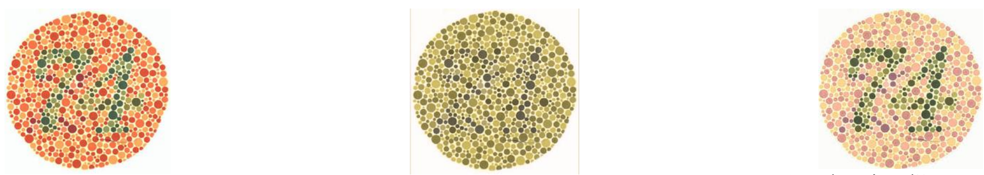
a) *Color-blindness*

As shown in table 17, ‘daltonism’ or color-blindness is more widespread that people usually imagine and affects 8% of men and 0.35% of women (DALTONIZE.ORG, 2013). There are also different types of color-blindness: Protanopia (lack of red photoreceptors, Deuteranopia (lack of green photoreceptors) and Tritanopia (lack of blue photoreceptors but is very rare). No cure exists but images can be adapted for these people according to the type of color-blindness.

Type of color vision deficiency	Prevalence in men (%)	Prevalence in women (%)
Protanopia	1	0.01
Protanomalous trichromatism	1	0.03
Deuteranopia	1	0.01
Deuteranomalous trichromatism	5	0.35
Tritanopia	rare	rare
Total prevalence	*8*	*0.4*

Table 17 – Types of blindness and repartition by gender (DALTONIZE.ORG, 2013)

In order to manage these deficiencies, applications have been developed to either help colorblind people to distinguish colors more efficiently or to allow regular people to see through colorblind people’s eyes. Such applications have been developed for iPhones or as an extension in the Google Chrome browser. Figure 29 shows an example:



Regular color

Simulated Pronatopia
(red deficiency)

“Daltonized” version
(adapted version)

Figure 29 – color-blindness example

Java Scripts also exist and can be implemented on websites in order to render colorblind adapted images or simulate color-blindness (see appendix). However, it is unlikely to see such function spread over all websites and the most convenient seems to emerge from browsers' settings and extensions, again. Indeed, including such feature on every website would increase codes to be downloaded and processed. It is therefore more convenient to activate this option on demand through applications or browsers' extensions.

b) *Blindness*

Even if some applications and websites integrate special features for users with partial sight through 'accessibility features', the majority does not and even less for severely impaired or blind people. In their researches, SIERRA and ROCA DE TOROGES (2012) propose a "Low Vision Mobile App Portal which provides a way to access mobile apps specifically designed for visually impaired users". These researches include specific interaction designs and would represent a whole thesis' subject. Only key findings are provided here.

As computers have become widely accessible, assistive technologies have been created to allow disabled people to access and use computers as everyone else (e.g. screen readers, speech recognition). With recent mobile technologies and the proliferation of touch screens, these assistive technologies also need to be adapted to meet users' expectations even with the lowest visual acuity. These adaptation features and designs have to be assessed as "despite the great effort of hardware manufacturers to include accessibility features in their touch based mobile devices, they are not good enough to obtain a good visually impaired user experience" (SIERRE & ROCA DE TOROGES, 2012). In order to be integrated on the Low Vision Mobile App Portal, applications must respond to specific requirements (i.e. adapted to low vision) while other features would be easily customizable by the user (e.g. icon size, screen contrast). There are plenty of advantages, from both sides. Developers reach a wider market while disabled people can access the portal and download adapted third parties applications as well as native applications.

On the design side, a key fact is that designs (e.g. menus, buttons) have to identical across applications. Besides these designs, easy control features have to be implemented such as a 'text to speech' library. Here are some examples of applications that have been customized for visually impaired users:

- A high contrast phone dialer (see figure 27);
- SMS, Email, Calendar and alarm apps are designed for low acuity users;

- Contacts and Battery apps use text to speech as well as special gestures.
- GPS apps give the user its location (City, Street and number).
- The camera from the phone makes easier for users to read books and watch images by inverting colors in order to increase the contrast.
- An API will soon be developed to allow developers to design their own applications and provide their own improvements.

All of this will make communications available to a wider proportion of the population and provides a bigger user experience to everyone.

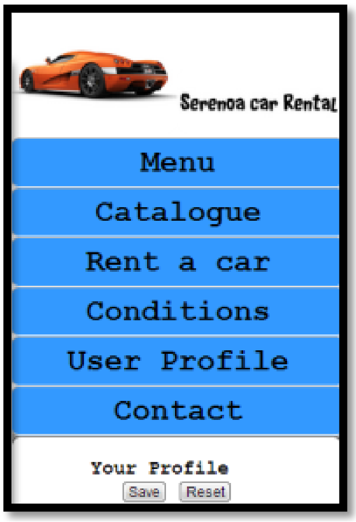
c) *Age & Tremor*

Taking users with hand tremor into account into the interfaces development also needs to go far beyond the scope of this thesis. However, key findings will be provided for those who want to go further on this topic. In a research conducted by WACHARAMANOTHAM et al. (2011) *"Evaluating Swabbing: a Touch-screen Input Method for Elderly Users with Tremor"*, researches provide the main following result: based on experimental researches, the study has confirmed that sliding/swabbing actions are more accurate (e.g. lower error rate) than tapping actions. Researches even provide useful thresholds for targets' width: tapping remain accurate if the target's width is higher than 54mm. Swabbing becomes more accurate under 41mm.

With the introduction of the 'Drag & Drop' feature in HTML5 (not integrated yet into mobile browsers (CANIUSE.COM, 2013), swabbing/sliding interaction designs could be implemented and increase satisfaction of elderly people with tremor.

Tremor statically appears within the sixth decennia (DEUSCHL, 2011). In that context convenient and costless tapping solutions may be provided if buttons are at least 40-50mm wide and if user's age is available. Similarly to an age-related customizable font size, interaction designs' could be adapted regarding users' age thanks to few JavaScript lines of code. Unfortunately, smallest smartphones are only 4cm wide on portrait mode. So buttons' width has to be set at 100%. Moreover, height can be configured to be higher and easily clickable as in the following example with smartphones:

```
JAVASCRIPT:
if (localStorage.age > 60) {
document.getElementById('text_button_1').style.lineHeight=2+(localStorage.a
ge/100)+'em';}
```



Rendering for user#1 (30years)
Smartphone
Orientatin: portrait
Buttons have normal size

Rendering for user#2 (70years)
Smartphone
Orientatin: portrait
Buttons are larger to avoid tapping's mistakes due to hands' tremor

5.3.1.4. Language, age, gender and current activity

As an ultimate feat, researchers aim at taking into account users' uniqueness. In this section, an overview is given for both 'fixed' characteristics (e.g. gender, language) and varying characteristics (e.g. current activity). Language feature has already been studied in the location dimension. Adaptations related to the age have been developed for sight's troubles and tremor issues.

Concerning font types and font colors, no researches have been conducted yet about gender, age and related preferred font types', font sizes' or color temperatures. Even if not relevant for the message, it should probably have an impact, at least from a marketing point of view. Indeed: "although findings are ambiguous, many investigations have indicated that there are differences between gender in preferences for colors" (KHOUW, 2005). Moreover, gender and age cannot be easily captured unless there is a user registration. An alternative (e.g. categorization) is presented in the next section.

In fact, user's current activity should be explained in the location dimension as the geolocation API may update the location several times at a specific interval to determine user's current main activity (e.g. walking, driving). The functioning is quite straightforward. At first, the time interval is set. Then, the information is sent to the Location Service server. Further, a list is sent back as a server response containing computed activities and their corresponding likelihood (see table 18 in appendix 2). Finally, according to developers' code, adaptations are made directly or the device asks the user to confirm which suggested activity is currently occurring. Table 18 (see appendix 2) provides the server's possible responses.

Once received, developers may interpret this kind of information and set adaptations techniques. For example, if someone is biking or running, text and buttons should be bigger while brightness and contrast should be increased to allow a better user experience.

5.3.2. Personality detection and users' categorization

A first simple example of categorization was the table presenting users' acuity impairment as they are aging. As the mobile first approach, considering that all users have a decreasing visual acuity as they are aging is beneficial for everyone' readiness. A second example of categorization was the assumptions that hands' tremor appears around 60 years. This section show how systems can capture and interpret parts of users' personality through the use of users' categorization applied to a *Safety Driving* application (NASOZ & LISETTI, 2007). This also shows that studying human personality interaction with systems in specific contexts is easier than in its global nature.

The safety driving application aims at performing "real-time emotion recognition and adapting the system to the affective state of the user depending on the user dependent specifics such as personality traits or users' preferences and the current context and application" (NASOZ & LISETTI, 2007). Building evolving user models is therefore fundamental as every user is different with evolving behaviors towards the system. As users evolve, systems have to be in a constant learning process in order to capture both preferences and personality. Here below is the procedure.

Within laboratories and through simulated reality applications, the first step is to extract, capture and analyze users' emotions in order to make a map of humans' physiological responses according to different emotions. Second step is to configure the system to the current emotional state by taking into account the surrounding environment as well as users' characteristics such as its personality. As a third step, the whole system is based on statistics and uses Bayesian Belief Networks in order to extract users' model and adapt itself to personal characteristics.

This method leads to the following results, showing that taking human behaviors into account gather a wide range of information such as the age, the gender and personality traits (NASOZ & LISETTI, 2007) as previous researches have proved they influence driving abilities: six main emotional states have been highlighted as influent while driving: "anger, frustration, panic, boredom, sleepiness, and non-negative"; five main traits of personality have been identified: "agreeable, conscientious, extravert, neurotic, open to experiments"; four age brackets: [0-25], [25-40], [40-60] and [60+]; the two genders [male, female] and current state of the car: damaged and not damaged. Decisions process is depicted in figure 28.

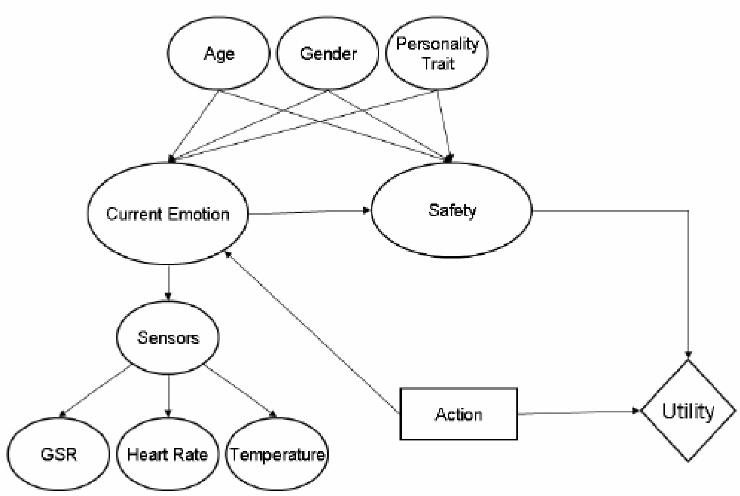


Figure 28 – Bayesian Belief Network representation of Driver Model (NASOZ & LISETTI, 2007)

According to the categorization made and the evaluated emotional state, following actions examples can be triggered or advices can be given: radio station switch, suggestion to stop/rest, suggestion to open the window, joke to calm the driver down.

Users' models are crucial in the field of adaptive user interactions. Unfortunately, such model is not complete as the lack of real date is consequent and makes the task harder. In this case, all the functional dependencies cannot be computed and further researches are necessary in collaboration with psychologists and transportations experts in order to evaluate these dependencies.

5.3.3. Key findings

Features	**Techniques**	**Pros**	**Cons**
Font type	Serif	Printed versions	Not legible on screens
	Sans-serif	Online supports. Headings for printed versions	Not legible enough on printed versions.
Font size	Fixed	No change in the layout	Users' settings are not taken into account
	Proportional to users' settings	Users' settings are taken into account	May affect the layout
	Categorization (age)	Assume sight's troubles at specific ages	Needs registration to extract users' age
Color Contrast	Black/white	Good comprehension and readiness	Not that attractive
	Low contrast or colors	Poor comprehension and readiness	More aesthetic
Color-blindness	javascripts	Colors' parameters may be customized. Works only if implemented on websites.	Time and resources consuming to implement and use. Only for images.
	Browser Extension	Easy to install and work on every website	Only for images. Not customizable.
Blindness	Specific applications	Increase blind users' experience with special features (text to speech).	Needs specific applications and features. Time and cost consuming.
Tremor	Categorization (age)	Adapts layout to hands' tremor and include other features (text to speech)	Needs registration to extract users' age. Swiping is preferred to tapping.
Current activity	Geolocation with intervals	Layout's adaptations according to the activity and the related level of attention	Needs geolocation enabled. Only determines the main activity (e.g. walking).
Personality	Categorization	Takes users' uniqueness into account.	Need deep researches. Easier to implement for a specific task (e.g. driving)

Table 24 – Key Findings for the user dimension and related adaptation techniques

Chapter VI: Conclusion & Discussion

On the next page, figure 30 sums up this work. On the one hand, adaptations techniques have been selected, gathered, tested, evaluated and ordered according to a suggested order of implementation that allows developers to structure websites' development by considering every context's dimensions and related impacts on users' experience. On the other end, users also have to set up some settings in order to allow adaptations regarding to their adaptation expectations as well as their privacy concerns. As a final discussion, advantages and shortcomings of such approach will be discussed. The focus will be at first set on the relevance of such adaptations in comparison with their actual impact on users' experience and their implementation costs. Moreover, users also have to make trade-offs between privacy issues and the expected level of adaptations. The more information developers have the more customizable but the more intrusive a system is. Finally, these shortcomings open doors for further researches.

6.1. Features diagram

As a conclusion, figure 30 and table 19 have been built. As depicted previously, the diagram advices to start implementing cross-platform features and then environments' and users' adaptations. Once again, it does not mean it is a serial process. It is just a path from the most general dimension to the most specific one and developers have to take into account next layers of adaptations while implementing a specific layer. These adaptations have been selected within the Serenoa Working area for their wide scope of application, their high level of relevance in many types of applications as well as the skills required to implement them. Some have been implemented, tested and evaluated while others, requiring time and high skills, have just been analyzed and evaluated thanks to previous researches. Every website does not have to include every feature and some dimensions may be emphasized in comparison to others depending on the website's purpose and developers' resources. This suggested order of implementation is in fact a web development tool to consider everything while creating information systems and websites. The path starts when developers begin implementing a website and the final goal is an increased user experience. On the way, developers choose adaptation techniques that will be displayed or used if users enable or disable some browser/device features. A plethora of combinations exists. On developers' side, HTML5 and CSS3 have been developed to make adaptive designs easier. Not using them would compromise the cross-platform adaptability. If no media queries are implemented, progressive enhancement is therefore not possible. On users' side, there are more constraints.

If the browser does not support HTML5 and CSS3, adaptive designs will not be properly displayed. Other things can be set up in users' browser or device and have important impacts on possible multidimensional adaptations (e.g. preferred font size, preferred language, geolocation and javascripts). However, these settings and features are normally enabled by default.

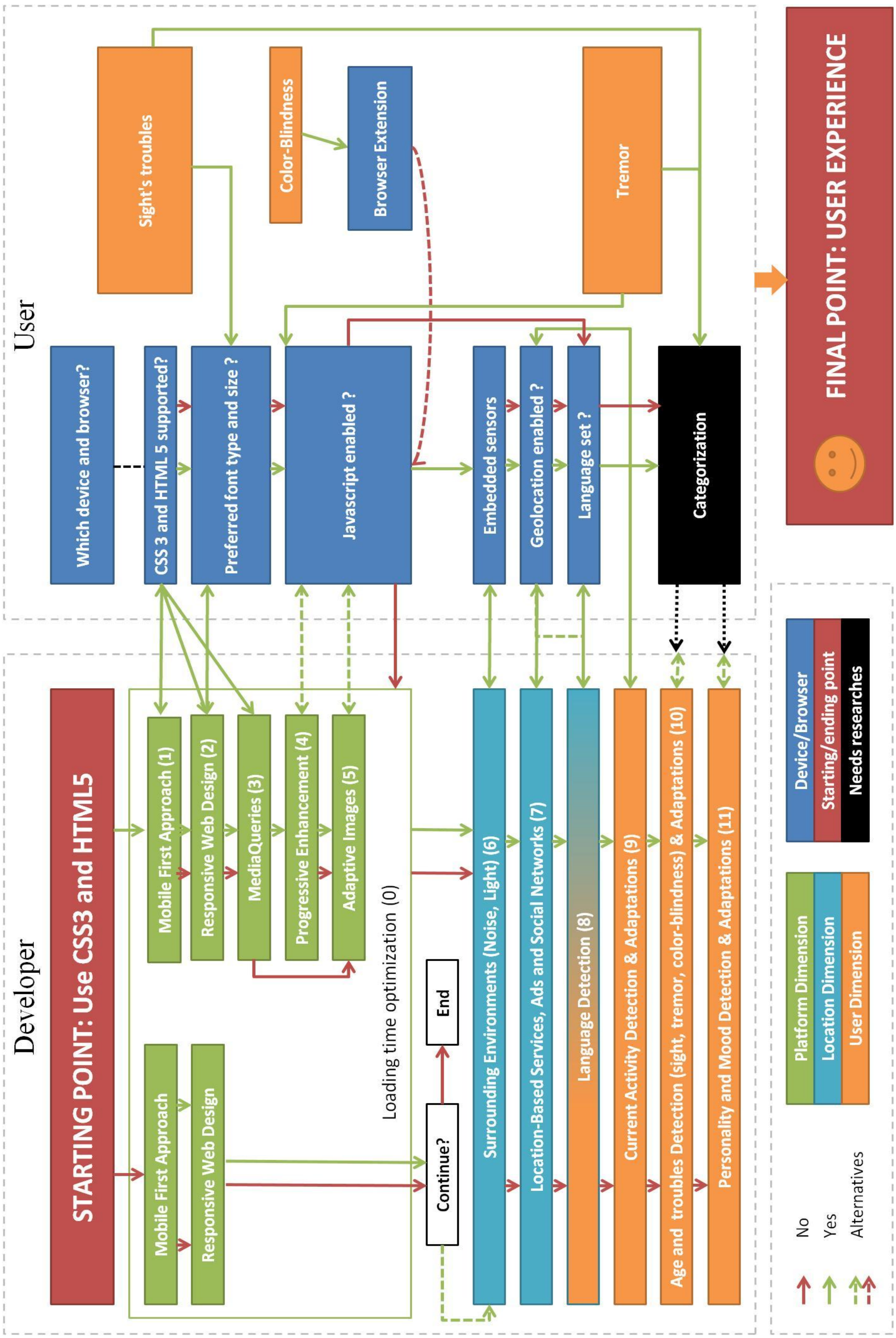


Figure 30 – Features' diagram

On developers' side, table 19 details development steps and techniques for each of them. On users' side, these adaptations will work if some device/browser features are enabled. Table 19 also highlights if the specific technique has been implemented on the car rental website or just analyzed. HTML, CSS and javascript files are available on the CD attached to this thesis. Pros and cons for each adaptation technique are provided for each dimension in the chapter 5 (5.1.3, 5.2.5 and 5.3.3)

Table 19 – Developers' choices and users' – Suggested order of implementation

Dimension	**Adaptation steps**	**Technique**	**Level of consideration**	**Impact on developers**	**User (it works if…)**
Platform	**Loading Time Optimization (0)**	Programming best practices	Implemented, tested and evaluated	Not time consuming if implemented since the beginning	
Platform	**Mobile First Approach (1)**	Buttons	Implemented, tested and evaluated	A bit longer than links to implement but fit on every platform	
Platform		Expandable contents	Implemented, tested and evaluated	To be implemented on smaller devices. Time consuming	javascript enabled
Platform		Drag and drop contents	Implemented, tested and evaluated	May be implemented for every device	javascript enabled HTML5 and CSS3 supported

Dimension	**Adaptation steps**	**Technique**	**Level of consideration**	**Impact on developers**	**User (it works if…)**
Platform	**Responsive Web Design (2)**	Proportional contents' widths	Implemented, tested and evaluated	Not time consuming if implemented since the beginning	HTML5 and CSS3 supported
Platform		Proportional font sizes	Implemented, tested and evaluated	Not time consuming	Preferred font size set in the browser's settings
Platform	**Media Queries (3)**	Setting several thresholds corresponding to different devices	Implemented, tested and evaluated	Develop Once, deploy Everywhere. Gain of time and money	HTML5 and CSS3 supported
Platform	**Progressive Enhancement (4)**	**Needs media queries** to add contents in function of different devices' capabilities	Analyzed and evaluated	Time and cost consuming But increases users' experience	javascript enabled (for the majority)
Platform	**Adaptive Images (5)**	javascripts to detect screen's width and send specific images accordingly	Implemented, tested and evaluated	Time consuming	javascript enabled (for the majority)

Dimension	**Adaptation steps**	**Technique**	**Level of consideration**	**Impact on developers**	**User (it works if…)**
Platform		New HTML5 element: <picture> (Does not exist yet)	Analyzed and evaluated	Time consuming to resize pictures several times but loading time improvements.	HTML5 and CSS3 supported
Environment	**Surrounding environments (6)**	Brightness regulation	Analyzed and evaluated (embedded into devices)	Not time consuming (embedded in devices)	Sensors activated and/or Application installed
Environment		Noise recognition	Analyzed (needs deep researches)	Deep researches and implementation	Microphone activated
Environment	**Location-based Services (7)**		Implemented, tested and evaluated	Time consuming but increases users' satisfaction sharply	Geolocation enabled

Dimension	**Adaptation steps**	**Technique**	**Level of consideration**	**Impact on developers**	**User (it works if…)**
Environment	**Language Detection (8)**	IP	Analyzed and evaluated	Implementation straightforward but still time consuming	No proxy, no VPN
Environment		HTTP Header	Analyzed and evaluated		Preferred language set
Environment		Geolocation API	Analyzed and evaluated		Geolocation enabled
Environment		Ask the user	Implemented, tested and evaluated		Confirmation
Environment		Regional websites	Analyzed and evaluated (time consuming)	Time consuming	No proxy, no VPN
User	**Current activity (9)**	Main activity deduced from users' speed	Analyzed and evaluated (time consuming)	Time consuming	Geolocation enabled
User	**Age & Sight's troubles (10)**	Font size adaptation	Implemented, tested and evaluated	Categorization (age) Time consuming	Preferred font size set or provide the age during registration

Dimension	Adaptation steps	Technique	Level of consideration	Impact on developers	User (it works if…)
User	**Age & Tremor (10)**	Tremor adaptation (interaction) Swiping instead of tapping.	Partially implemented (buttons), tested and evaluated	Categorization (age) if age not provided. Adaptations if age provided. Time consuming	Shares age during registration
User	**Color-blindness (10)**	javascripts	Implemented, tested and evaluated	Time consuming	javascript enabled
User		Browser's extension	Analyzed, installed, tested and evaluated	May recommend the extension	Extension installed
User	**Blindness (10)**	Rethink and reshape applications	Analyzed and evaluated (time consuming)	Time consuming for specific applications	
User	**Personality & Mood (11)**	Categorization	Analyzed and evaluated (time consuming)	Time consuming and specific researches needed	Agreement (privacy issues)

6.2. Advantages and shortcomings

The main advantage of this research is its wide scope even if limited to websites 'main features, graphical interfaces and interaction designs. Indeed, while some adaptations have been tested and evaluated, some more requiring more time and skills have been illustrated by previous or current researches. This allows the reader to have a deep multidimensional overview and an evaluation of specific adaptation techniques. If necessary, readers may easily access knowledge resources to go deeper in specific field. This thesis may easily be used for any information systems' development in order to structure the process and focus on some features. Nevertheless, some aspects of information systems' development process have not been studied such as the relevance of these adaptations in comparison with development's costs and effects on users' experience as well as rising users' privacy concerns. Finally, researches focused only on the three current main platforms (i.e. computers, tablets and smartphones) and on specific amount of specific adaptation techniques. However, there is a plethora of other adaptation techniques (around 150 on the Serenoa Working are) and devices with specific features exist or will appear soon (e.g. smart TV, Google glass).

6.2.1. Relevance of these adaptations

As depicted in the table, most of these adaptations are time consuming from a developer's point of view. And as time means money, the relevance of these adaptations have to be addressed. Indeed, even if the developers' community recognizes context-aware adaptations and their related models provide several benefits (e.g. lower development costs and length, better users' interaction and experience), related costs seem to surpass benefits and these adaptations are therefore partially adopted. Following arguments are given: "a steep learning curve is required to understand its concepts, to use it, to apply it in a large scale, additional phases must be added to the development process, more resources are needed, and so on" (MOTTI et al., 2013). As a result, providing developers' community with simple models and tools would be a step forward towards a wider adoption of context-aware adaptations. Moreover, as in every sector, changes will come from the demand side. If clients are aware of such possibilities, pressure will be put on researchers and developers to integrate and develop such features. It is from this perspective that this thesis has been written. It offers the reader a wide overview of adaptations techniques and their analysis which provides finally a tool for the creation of context-aware websites and applications. From a user's point of view, relevance of such adaptations may also be questioned. As users perceive things before interpreting or evaluating them, layouts' adaptations seem to be more relevant than other features (NORMAN, 1988b) which can be time and resources consuming for developers.

6.2.2. Privacy issues

The large adoption of these recent technologies also raises users' concerns about their privacy. A trade-off is then necessary between features' usefulness and users' privacy. Indeed, most developed devices and applications would know exactly users' location, mood, activities, who is around and most users perceive these technologies as very intrusive and do not want to feel tracked. Researches pointed out that people are generally not likely to allow an application to track their position unless the service is useful (BARKUUS & DEY, 2003). Therefore developers have to build a trustworthy relationship with users by using their location properly with their express permission and not sharing it with third parties.

6.3. Further work

This thesis provides a useful web development tool which provides a global evaluation of selected multidimensional adaptations techniques as well as a set of recommendations and guidelines to increase users' experience. However, taking context's dimensions into account transforms previous development processes and a costs' analysis would be therefore necessary. Beyond cross-platform portability and related costs, surveys would be useful to know what users expect in the near future regarding context's adaptations. Moreover, despite the Location Privacy Protection Act introduced in 2012, deeper researches on users' privacy protection and data legislation would have to be conducted in order to produce a strong legislative framework on the subject. This would make applications and websites more reliable and users would be more eager to share personal data. Finally, other adaptation techniques have to be evaluated on a wider range of devices.

Bibliography

ADOBE (2010). *Adobe Mobile Experience Survey: What Users Want from Media, Finance, Travel & Shopping.* http://www.keynote.com/docs/news/AdobeScene7_MobileConsumerSurvey.pdf (Retriever June 10, 2013)

ANDROID (2013). *Recognizing the User's Current Activity*. http://developer.android.com/training/location/activity-recognition.html (retrieved July 20, 2013)

ARDITI, A. (2013). *Making Text Legible*. http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible (retrieved June 5, 2013)

ATKINSON, P. (2005). *Man in a briefcase: the social construction of the laptop computer and the emergence of a type form.* Journal of Design History, 18 (2), 191-205. doi: 10.1093/jdh/epi024.

BARKUUS L., DEY A. (2003). *Location-Based Services for Mobile Telephony: a Study of Users'.* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.527&rep=rep1&type=pdf (retrieved July 15, 2013)

BASTIEN J.M.C., SCAPIN, D.L. (1993). Ergonomic Criteria for the Evaluation of Human-Computer Interfaces. *Inria, n°156.* oai: hal.inria.fr:inria-00070012. http://hal.inria.fr/docs/00/07/00/12/PDF/RT-0156.pdf (retrieved June 17, 2013)

BBC (2005). *Tracking a suspect by mobile phone.* http://news.bbc.co.uk/2/hi/technology/4738219.stm (retrieved July 25, 2013)

BERGMAN, E. (2000). *Information Appliances and Beyond: Interaction Design for Consumer Products (Interactive Technologies).* Bergman, E. (ed.). US: Morgan Kaufmann. http://books.google.be/books?id=wtPlin2FNXMC&printsec=frontcover&hl=fr&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false (retrieved June 25, 2013)

BRUSILOVSKY, P. (1996) Methods and techniques of adaptive hypermedia. In *User Modeling and User-Adapted Interaction (UMUAI)*, vol. 6 (2-3), 87-129. doi: 10.1007/BF00143964. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.7794&rep=rep1&type=pdf (retrieved June 2, 2013)

BRUSILOVSKY, P. (1996b). Adaptive hypermedia, an attempt to analyze and generalize. In P. Brusilovsky, P. Kommers, & N. Streitz (Eds.), *Multimedia, Hypermedia, and Virtual Reality*. Vol. 1077, 288-304. Berlin: Springer-Verlag. http://rd.springer.com/book/10.1007/3-540-61282-3/page/2# (retrieved June 10, 2013)

BUSINESS INSIDER (2013). *How Location-Based Services Are Transforming The Mobile Industry*. http://www.businessinsider.com/location-data-is-transforming-mobile-2013-4 (retrieved July 1, 2013)

BUXTON, B. (2001). *Less is More (More or Less): Uncommon Sense and the Design of Computers*. In P. J. Denning (Ed.). *The Invisible Future: The Seamless Integration of Technology Into Everyday Life* (pp.145-179), US: McGraw-Hill Companies Caniuse.

CHELARIU, R. (2013*). Responsive Web Design With Physical Units*. http://mobile.smashingmagazine.com/2013/03/21/responsive-web-design-with-physical-units/ (retrieved June 15, 2013)

CHEN, B. X. (2010). What the iPad means for the future of computing. http://www.wired.com/gadgetlab/2010/02/ipad-future/ (retrieved June 12, 2013)

CHENG, R. (2013). Tablets expected to surpass desktop, laptop sales by 2015. http://news.cnet.com/8301-1035_3-57577905-94/tablets-expected-to-surpass-desktop-laptop-sales-by-2015/ (retrieved June 27, 2013)

CHU S., NARAYANAN S., KUO J. (2006). *Content analysis for acoustic environment classification in mobile robots.* http://sail.usc.edu/aigaion2/index.php/publications/show/161 (retrieved June 29, 2013)

CISCO (2013). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017.* http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf (retrieved June 12, 2013)

COMPUWARE (2011). *What Users Want from Mobile.* http://e-commercefacts.com/research/2011/07/what-usrs-want-from-mobil/19986_WhatMobileUsersWant_Wp.pdf (retrieved June 12, 2013)

COMSCORE DATA MINE (2013). Smartphones Reach Majority in all EU5 Countries. http://www.comscoredatamine.com/2013/03/smartphones-reach-majority-in-all-eu5-countries/ (retrieved June 25, 2013)

DALTONIZE.ORG (2013). *There is not just one color blindness.* http://www.daltonize.org/search/label/Color%20Blindness (retrieved June 27, 2013)

DALTONIZE.ORG, (2013). *There is not just one color blindness*. http://www.daltonize.org/search/label/Color%20Blindness (retrieved July 5, 2013)

DAY, A. K. (2001). Understanding and Using Context. *In Personal and Ubiquitous Computing*, 5 (1), 4-7. doi: 10.1007/s007790170019.

DE KOCH, N. (2001) *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Proces*s. UNI-DRUCK Verlag, 2001. PhD. Thesis, München: Ludwig-Maximilians-Universität. Online Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process

DEUSCHL G. (2011). *Tremblement essentiel héréditaire*. http://www.orpha.net/consor/cgi-bin/Disease_Search.php?lng=FR&data_id=231&Disease_Disease_Search_diseaseGroup=tremor&Disease_Disease_Search_diseaseType=Pat&Disease(s)/group%20of%20diseases=Hereditary-essential-tremor&title=Hereditary-essential-tremor&search=Disease_Search_Simple (retrieved June 18, 2013)

DISTIMO (2013). New Study: Apple's App Store Crushing Google Play in Revenue. http://www.patentlyapple.com/patently-apple/2013/05/new-study-apples-app-store-crushing-google-play-in-revenue.html (retrieved June 15, 2013)

DOURISH, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1), 19-30. doi: 10.1007/s00779-003-0253-8.

EQUATION RESEARCH (2009). *Why the Mobile Web is Disappointing End-Users.* A study *of consumers' mobile web experiences conducted by Equation Research on behalf of Gomez, Inc.* http://www.londonwebperformance.org.uk/component/content/article/1-wpo-knowledgebase/121-gomez-mobile-web-user-experience-survey-why-the-mobile-web-is-disappointing-end-users- (retrieved June 15, 2013)

FROST, B. (2011). *What Is Mobile-First Responsive Web Design?* http://bradfrostweb.com/blog/web/mobile-first-responsive-web-design/ (rertrieved June 27, 2013)

GARTNER (2013). Press release Gartner Says PC Market in Western Europe Declined 20.5 Percent in First Quarter of 2013. http://www.gartner.com/newsroom/id/2484815 (retrieved June 29, 2013)

GARTNER (2013b). Press release Gartner Says Worldwide PC, Tablet and Mobile Phone CombinedShipments to Reach 2.4 Billion Units in 2013. http://www.gartner.com/newsroom/id/2408515 (retrieved June 12, 2013)

GARTNER REPORT (2013). Press release Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012. http://www.gartner.com/newsroom/id/2335616 (retrieved June 30, 2013)

GRISBY, J. (2010). CSS Media Query for Mobile is Fool's Gold, Online on the site Cloud for Blog. http://blog.cloudfour.com/css-media-query-for-mobile-is-fools-gold/ (retrieved June 27, 2013)

GUEMBEL (2012). *Geolocation and Language Detection: A Primer.* http://www.global-sites.com/en/geolocation-language-detection-primer/ (retrieved July 10, 2013)

HALLECK, Th. (2013). Google Play Catching Up To Apple In Downloads But iOS App Store Still Tops In Revenue. Online on the site International Business Time http://www.ibtimes.com/google-play-catching-apple-downloads-ios-app-store-still-tops-revenue-1289289 (retrieved July 15, 2013)

HEALTH, A. (2009). *About progress from one-size-fits-all to personalization*. Online on the site Axelrod Access for All http://axelafa.com/about (retrieved June 2, 2013)

HEITING, G. (2010). *How Your Vision Changes as You Age*. Online on All about Vision.com http://www.allaboutvision.com/over60/vision-changes.htm. (retrieved June 21, 2013)

IDC, (2013). *Android and iOS Combine for 92.3% of All Smartphone Operating System Shipments in the First Quarter While Windows Phone Leapfrogs BlackBerry, According to IDC.* Press Release.http://www.idc.com/getdoc.jsp?containerId=prUS24108913 (retrieved June s12, 2013)

IDC, (2013b). Press Release *Worldwide Tablet Market Surges Ahead on Strong First Quarter Sales.* http://www.idc.com/getdoc.jsp?containerId=prUS24093213 (retrieved June 7, 2013)

I-SKOOL, Interactive learning (2012). *Web Design for Mobiles and Tablets – Viewport Sizes*. http://i-skool.co.uk/mobile-development/web-design-for-mobiles-and-tablets-viewport-sizes/ (retrieved June 28, 2013)

JOHANSSON, J. (2013). *How To Make Your Websites Faster On Mobile Devices*. mobile.smashingmagazine.com http://mobile.smashingmagazine.com/2013/04/03/build-fast-loading-mobile-website/ (retrieved June 21, 2013)

KAMEKA, A. (2013) *Apple iOS app store reaches 900,000 apps, 375,000 iPad-optimized*. Online on mobileburn.com http://www.mobileburn.com/21702/news/apple-ios-app-store-reaches-900000-apps-375000-ipad-optimized#mkGHRL6JWdWFEbZr.99.

KENDO UI/TELERIK. (2012). *HTML5 Adoption Fact or Fictio*n. Online http://www.kendoui.com/docs/default-source/surveys-docs/html5_adoption_survey.pdf?sfvrsn=2.

KEYNOTE (2012). *2012 Mobile User Survey - Mobile User Preferences, Habits, and Expectations.* Online http://www.keynote.com/docs/reports/Keynote-2012-Mobile-User-Survey.pdf.

KHOUW N. (1995). *The meaning of color for gender.* http://www.colormatters.com/khouw.html (retrieved July 4, 2013)

KJELDSKOV, J. (2013). Mobile Computing. In Soegaard, Mads & Dam, Rikke Friis (Eds.). *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* Chapter 9. Aarhus, Denmark: Interaction Design Foundation. Online http://www.interaction-design.org/encyclopedia/mobile_computing.html.

KOSTAKOS, V., NICOLAI, T., YONEKI, E., O'NEILL, E., KENN, H. & CROWCROFT, J. (2009). *Understanding and measuring the urban pervasive infrastructure. Personal and Ubiquitous Computing,* 13(5), 355-364. London: Springer-Verslag. doi: 10.1007/s00779-008-0196-1.

LINDHOLM, Chr., KEINONEN & KILJANDER, T. (2003). *Mobile Usability: How Nokia Changed the Face of the Mobile Phone.* US: McGraw-Hill Professional.

LOPEZ-JAQUERO V., VANDERDONCKT J., MONTERO F., GONZALEZ P. (2008). *Towards an Extended Model of User Interface. Adaptation: the ISATINE framework.* http://www.dsi.uclm.es/personal/VictorManuelLopez/mipagina/archivos/Lopez-EIS2007 (retrieved July 27, 2013)

MARTIN, M. (2009). *Typographic Design Patterns and Best Practices*. smashingmagazine.com http://www.smashingmagazine.com/2009/08/20/typographic-design-survey-best-practices-from-the-best-blogs/ (retrieved July 14, 2013)

MC CULLOUGH, M. (2004). *Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing.* US: The MIT Press.

MITROVIC, R., ROYO, J.A. & MENA, E. (2005). *Performance Analysis of an Adaptive User Interface System Based on Mobile Agent.* http://zaguan.unizar.es/record/1990/files/INPRO--2009-028.pdf (retrieved June 30, 2013)

MOGGRIDGE, B. (2007). *Designing Interactions*. US: The MIT Press.

MOTTI V.G., RAGGETT D., VANDERDONCKT J. (2013). *Current Practices on Model-based Context-aware adaptation.* http://ceur-ws.org/Vol-1013/casfe2013_submission_4.pdf (retrieved July 20, 2013)

MURPHY, J., KJELDSKOV, J., HOWARD, S., SHANKS, G. & HARTNELL-YOUNG, E. (2005): The converged appliance: "I love it... but I hate it". In Proceedings of OZCHI, *05, the 17th Annual Conference on Human-Computer Interaction*, 1-10. Australia: Computer-Human Interaction Special Interest Group (CHISIG) of Australia Narrabundah.

NASOZ F., LISETTI C. (2007). Affective User Modeling for Adaptive Intelligent User Interfaces. http://dl.acm.org/citation.cfm?id=1769590.1769637 (retrieved June 12, 2013)

NATIONAL EYE INSTITUTE (2013). *Facts About Dry Eye.* http://www.nei.nih.gov/health/dryeye/dryeye.asp#7 (retrieved June 22, 2013)

NEBELING M., NORRIE M. & MATULIC, F. (2013). *Beyond Responsive Design.* http://www.globis.ethz.ch/beyondresponsive.pdf (retrieved June 11, 2013)

NEBELING, M. & NORRIE M. C. (2013). *Responsive Design and Development: Methods, Technologies and Current Issues.* http://www.globis.ethz.ch/icwe2013/ (retrieved June 19, 2013)

NETMARKETSHARE (2013). *Mobile/Tablet Browser Market Share* http://www.netmarketshare.com/browser-market-share.aspx?qprid=0&qpcustomd=1&qpcd=1300 (retrieved June 17, 2013)

NIXON, R. (2011). *Html5 Crash Course: Learn Html5 in 20 Easy Lectures,* Paperback, 316 pages. UK Nixon Publishing.

NORCIO, A.F. & STANLEY, J. (1989). Adaptive Human-Computer Interfaces: A literature survey and perspectives. *IEEE Transactions on Systems, Man and Cybernetics,* 19(2), 399-408. doi: 10.1109/21.31042.

NORMAN, D. A. (1998).*The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex and Information Appliances Are the Solution.* US: The MIT Press.

NORMAN, D. A. (1998b). *The Psychology of Everyday Things.* US: The MIT Press.

RAMSAY, M. & NIELSEN, J. (2000). *WAP Usability. Report from a Field Study in London, Fall 2000 Déjà Vu: 1994 All Over Again*. Online http://www.nngroup.com/reports/wap/WAP_usability.pdf (retrieved July 15, 2013)

RESPONSIVEIMAGES IMAGES COMMUNITY GROUP (2013). S*ample markup for `picture`*. http://responsiveimages.org/ (retrieved July 1, 2013)

ROGERS, Y., SHARP, H. & PREECE, J. (2011). Interaction Design: Beyond Human-Computer Interaction (3rd ed.). UK: John Wiley and Sons. Online Limited version http://books.google.be/books?id=b-v_6BeCwwQC&printsec=frontcover&hl=fr#v=onepage&q&f=false (retrieved June 30, 2013)

SARAN, C. (2013). *Gartner predicts the demise of the desktop PC refresh cycle.* Online on the site Computerweekly.com http://www.computerweekly.com/news/2240176063/Gartner-predicts-the-demise-of-the-desktop-PC-refresh-cycle (retrieved June 30, 2013)

SCHILIT, B. N. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8 (5), 22-32. doi: 10.1109/65.313011.

SCHMIDT, ALBRECHT, BEIGL, MICHAEL, GELLERSEN, HANS-WERNER (1999). *There is more to context than location*. In Computers & Graphics, 23 (6) 893-901.

SERRA, M. (2012). *Pas de Safari 6 pour les utilisateurs Windows.* Online on the site C/net France http://www.cnetfrance.fr/news/pas-de-safari-6-pour-les-utilisateurs-windows-39774596.htm.

SIERRA J.S, ROCA DE TOROGES J.S (2012). *Designing Mobile Apps for Visually Impaired and Blind Users.* http://www.techrepublic.com/resource-library/whitepapers/designing-mobile-apps-for-visually-impaired-and-blind-users/ (retrieved July 1, 2013)

SMITH, M. (2012). *Google Chrome browser arrives on Android (video)*. Online http://www.engadget.com/2012/02/07/google-chrome-browser-arrives-on-android-video/.

SULLIVAN, M. (2012). *3G/4G Performance Map: Data Speeds for AT&T, Sprint, T-Mobile, and Verizon*. Online http://www.techhive.com/article/254888/3g_4g_performance_map_data_speeds_for_atandt_sprint_t_mobile_and_verizon.html (retrieved June 3, 2013)

VAN GROVE (2010). *Top 5 Location-Based Services.* http://mashable.com/2010/10/27/best-location-based-services/ (retrieved July 2, 2013)

W3.ORG (2013). *Geolocation API Specification*. http://www.w3.org/TR/geolocation-API/#api_description (retrieved June 23, 2013)

W3C (2013). *Members' list.* http://www.w3.org/Consortium/Member/List (retrieved June 18, 2013)

W3SCHOOLS. (2013). *OS Platform statistics*. http://www.w3schools.com/browsers/browsers_os.asp (retrieved June 4, 2013)

W3SCHOOLS.com. (2012). *Media Queries : W3C Recommendation*. http://www.w3.org/TR/css3-mediaqueries/. (retrieved June 10, 2013)

W3TECHS.COM (2013). *Usage statistics and market share of Apache for websites*. Online http://w3techs.com/technologies/details/ws-apache/all/all (retrieved June 17, 2013)

WACHARAMANOTHAM C., HURTMANNS J., MERTENS A., KRONENBUERGER M., SCHLICK C., BORCHERS J. (2011). *Evaluating Swabbing: a Touchscreen Input Method for Elderly Users with Tremor*. http://dl.acm.org/citation.cfm?id=1979031 (retrieved June 12, 2013)

WEB BUSINESS (2012). *Location Based Marketing – LBS, NFC, Bluetooth and LBA.* http://webbusiness2go.com.au/location-based-marketing/ (retrieved July 15, 2013)

WHEILDON, C. (2005). *Type & Layout: Are You Communicating or Just Making Pretty Shapes*. http://www.amazon.com/Type-Layout-Communicating-Making-Pretty/dp/1875750223/ref=sr_1_1/104-2853119-5415149?ie=UTF8&s=books&qid=1176554144&sr=8-1 (retrieved June 21, 2013)

WHITNEY, L. (2011). *HTML5-enabled phones to hit 1 billion in sales in 2013*. http://news.cnet.com/8301-1023_3-57339156-93/html5-enabled-phones-to-hit-1-billion-in-sales-in-2013/ (retrieved June 15, 2013)

WILCOX, M. (2013). *Adaptive Images.* http://adaptive-images.com/ (retrieved June 29, 2013)

WINDOWS. (2010). *DPI and Device-Independent Pixels*. http://msdn.microsoft.com/en-us/library/windows/desktop/ff684173(v=vs.85).aspx (retrieved June 9, 2013)

WOOD, J. (2011). *The best fonts to use in Print, Online ans Email*. Online http://www.awaionline.com/2011/10/the-best-fonts-to-use-in-print-online-and-email/ (retrieved June 6, 2013)

WROBLEWSKI, L. (2013). *New Layouts for the Multi-Device Web.* Online http://www.lukew.com/ff/entry.asp?1721 (retrieved June 16, 2013)

ZICKUHR K. (2012). *Three-quarters of smartphone owners use location-based services.* http://pewinternet.org/Reports/2012/Location-based-services.aspx (retrieved July 7, 2013)